

## Open Source

Автор — Sean Lamb.

Автор перевода — Д.Семенов

*Одна из огромных достоинств OPEN SOURCE — собственно говоря в том, что вам доступен исходный код фактически любой программы. Данная статья обсуждает проблемы которые могут возникнуть у вас, когда вы пытаетесь установить программу, используя ее исходники. В общем-то у Linux'a есть хорошие встроенные возможности по установке программ из пакетов, таких как rpm в RedHat. Если вы уже устанавливали программы, с использованием RPM или еще-чего-нибудь, то вам будет легче понять обсуждаемые здесь вещи.*

### Собственно source

Первое, что вам нужно — это естественно исходники той программы, которую вы собираетесь установить. Программу для установки вы можете утянуть с любого FTP. В данной статье мы рассмотрим установку двух программ — KDevelop и klavg (показывает загрузку системы в KPanel), обе программы доступны для скачивания по адресу: KDE applications page (<http://www.kde.org/applications.html>).

На моей собственной системе я использую отдельную директорию для хранения всех скачанных файлов. Все скачанные файлы находятся в директории — ~/download (так как я захожу как простой пользователь, а не как root<sup>1</sup>). Помещение всех файлов в один общий каталог преследует две цели — я всегда помню где находится скачанный мною файл, кроме того если однажды я пытаюсь скачать снова файл, который уже был мною скачан, то мое программное обеспечение скажет мне, что такой файл уже имеется в каталоге и я не буду попусту тратить время на бесполезную закачку.

### \*.TAR

По большей части исходники программ поставляются как файлы ".tar" — так называемые "tarballs". Это по своей сути означает, что все файлы, которые нужны для инсталляции выбранной вами программы находятся в одном архивном файле, созданным при помощи программы tar. Поскольку сама программа tar не производит упаковку, то бишь сжатие файлов, то иногда такой файл прогоняют через упаковщик — либо GZIP, либо BZIP. В таких случаях расширения файла становится tgz.

Итак следующим нашим шагом будет распаковка архива. Поскольку большинство дистрибутивов поставляется с KDE или GNOME, в качестве оконного менеджера — большинство Linux пользователей имеют установленный пакет — типа нашего RAR'a. Для распаковки архива из командной строки просто наберите следующий код:

```
tar zxvf kdevelop-1.0beta2.tar.gz
```

Данная команда распакует все файлы в текущий каталог, создаст необходимые каталоги (как они определены в tar файле).

Я использую KDE в качестве своего рабочего стола, то бишь мы будем рассматривать необходимые для нашей работы утилиты, поставляемые с KDE. Аналогичные утилиты есть и в GNOME.

Для извлечения файла из архива щелкаете по файлу в KFM и дальше — Archiver. Из Archiver Edit меню выберите Extract, щелкните All Files и выберите директорию — куда скидывать файлы. Для KDevelop мы видим, что первая запись в tarball'e =

```
drwxrwxrwx 504/100 0 1998-08-22 kdevelop-1.0beta2/
```

Таким образом данный архив распакуется внутри указанной вами директории, в поддиректории с именем kdevelop-1.0beta2.

### R is for Readme

Следующий шаг, после того как архив распакован в нужную директорию заключается в том, что вам нужно прочитать файл Readme. Данный файл содержит важную информацию о том, какое программное обеспечение необходимо для того чтобы программа работала после установки.

Часто данный файл содержит пошаговые инструкции по установке программы. Если Tarball файл не содержит данного файла Readme то поищите файл аналогичного назначения, называемый Install. Для Kdevelop файл Readme отсылает user'a к файлу INSTALL. Который содержит команды, необходимые для инсталляции программы:

```
./configure
make
make install
```

README который включен вместе с klavg не описывает шаги для установки, и вообще такого файла в tarbal'e вы не найдете. На домашней страничке программы сказано, что для того, чтобы установить программу — вам будет нужно ввести лишь 2 команды:

```
make
make install
```

Почти для всех программ инсталляционный процесс как правило одинаков. Существуют только три необходимые команды — (./configure, make and make install) — и больше кроме них вам ничего не нужно. Первая команда — ./configure — запускает последовательность shell скриптов для определения местоположения необходимых библиотек и командные имена компиляторов и линковщиков. Эта же команда создает файл Makefile, который используется для запуска компилятора и линкера, а также для копирования скомпилированных бинарников в новое местоположение.

Откройте сейчас терминальное окно, и CD в директорию куда вы распаковали файлы программы. В моем случае это /home/slambo/download/kdevelop-1.0beta2/ (/home/slambo/download/klavg/ — для klavg). Далее запустите первую команду — ./configure. На экране вы увидите примерно следующее:

```
[slambo@jenlamb kdevelop-1.0beta2]$ ./configure
creating cache ./config.cache
checking for extra includes... no
checking for extra libs... no
checking for a BSD compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking whether make sets ${MAKE}... yes
checking for working aclocal... found
checking for working autoconf... found
checking for working automake... found
checking for working autoheader... found
checking for working makeinfo... found
checking for a C-Compiler...
checking for gcc... gcc
...
updating cache ./config.cache
creating ./config.status
creating Makefile
creating kdevelop/Makefile
creating po/Makefile
creating kdlgloder/Makefile
...
creating config.h
[slambo@jenlamb kdevelop-1.0beta2]$
```

Сейчас я вам показал, только лишь малую долю того, что у вас будет выводиться на экран. На самом деле будет выведено гораздо больше информации. Первая линия — "creating cache ./config.cache" — создает временный файл.

Команд make делает две операции:

- Ø Компиляция
- Ø Линковка (linking)

Как правило если на этом этапе вам выдается какая-либо ошибка то это как правило говорит о том, что у вас нет каких-либо необходимых библиотек. Если это так — то вам нужно будет либо установить либо upgread'ить старые версии. После этого попытайтесь заново установить программу.

## **Read The Fantastic [sic] Manual(RTFM)**

Итак если команда `./configure` завершилась успешно, но на этапе компиляции вы получили ошибку, не связанную с отсутствием какой-либо библиотеки...что делать? Первое — это читать RTFM. Проверьте документацию, которая идет вместе с исходным кодом. Иногда некоторые примечания даются в файлах `Readme` или `Install`, но как правило их там мало, либо они очень неинформативные. Итак если предоставляемая вами документация не помогла решить вам вану проблему, вы можете пойти на сайт программы — там могут быть `patch`'и и `update`'ы для вашей программы. Если и это вам не помогло — попробуйте поискать ответы путем поиска по конференциям Usenet через систему DeJa.com. Используйте сообщение об ошибке в качестве текста для поиска.

Если и этот вариант вам ничем не помог — вы можете напрямую обратиться к разработчику программы. Но как правило до этой стадии доходит редко...Порывшись хорошенько в инете — вы сможете найти ответ на любой ваш вопрос. Главное хорошо искать.

## **"God, Root, what is difference?"**

Вообще-то мы пропустили одну важную деталь. Прежде чем выполнить третью команду, вы либо должны пользователю дать соответ-ие привелегии на команду установки и необходимые директории, либо вы должны зайти как `root` в систему. Иначе при попытке установки программы вы можете получить сообщение `{Permission ...}` что говорит о вашей невозможности выполнить команду — у вас нет привилегий, которые может вам дать только админ системы.

Не буду объяснять, но скажу что нельзя давать простым юзерам заходить в систему как `root` — это может быть большим риском для системы. Посему `root`-пароль не должен никому разглашаться.

Зайти как `root` в систему вы можете при помощи команды `su` (это в том случае если вы законнектились в систему, как обычный `user`). Прочитайте `man` по этой команде. Теперь смело запускайте команду `make install`.

Если на этом этапе вы получаете сообщение об ошибке. То как правило это означает, что файлы копируются (пытаются)в несуществующую директорию.

Как только эта команда завершилась делаете `logout` и пробуете новую прогу. Итак вверху находятся две картинки после того как я установил `Kdevelop` и `Klavg`.

## **Итак, где же еще могут быть грабли?**

Давайте представим ситуацию, что содержимое файлов `Readme` и `Install` не сильно вам помогают понять что же нужно сделать. Что тогда? В принципе, команды для компиляции и инсталляции всегда одни и те же. Если в файле `tar` имеется файл `makefile` то вы можете использовать команду `make` для компиляции программы, и вероятно для ее установки. Если в `tar` файле также имеется и `config` скрипт, то запустите его первым — перед запуском команды `make`.

## **Примечания**

[1] Вход в систему как обычный пользователь, нежели как `root`, помогает вам избежать множество неприятных моментов. В частности вы сможете протестировать программу на предмет доступа к ней обычных пользователей системы, а не `root`'ских пользователей.

[2] Последние версии `winzip`'а позволяют работать с `tgz` файлами, как с обычными `zip`'ими архивами.

## **The Linux Ukrainian HOWTO**

### **Не очень короткое предисловие...**

На этой странице я попытаюсь осветить все проблемы, которые приходится решать при украинизации (точнее — локализации) Линукса. Все настройки описываются для `RedHat 4.x`, но особых проблем при применении их для других поставок не должно быть. Конечно, решить любую проблему можно несколькими различными путями, но на данной странице мало внимания уделяется всем возможным вариантам. Так что если кто-то знает какие-нибудь тонкости и

хитрости украинизации различных программ, я с радостью выслушаю их и опишу здесь для общего ознакомления.

Очень рекомендую почитать The Linux Cyrillic HOWTO<sup>1</sup> в изложении Alexander L. Belikoff. Также советую сходить на KOI8-R References<sup>2</sup> (Russian Net Character Set) Андрея Чернова.

К сожалению, в момент создания данной страницы rfc на koi8-u еще не принят, поэтому все нижеизложенное основывается на koi8-u draft.

В будущем будет предпринята попытка создать "easy to install package", содержащий все необходимые файлы и утилиты для украинизации. И конечно же я буду рад любым замечаниям, пожеланиям и особенно похвалам.

Так что не забывайте сюда заглядывать, может что-то будет меняться...

**[21.04.98]** rfc уже приняли! Его номер — rfc 2319<sup>3</sup>. С чем всех и поздравляю!

**[05.08.98]** В связи с тем, что я, наконец-то, получил возможность поиздеваться над glibc-версией Linux-а, а именно RedHat 5.1, я добавил некоторую информацию, связанную с различиями в украинизации libc-5 и glibc-2 поставками системы. Эти различия проявились пока только в Локализации. Также были убраны «лишние» мелькающие иконки, т.к. страница выглядела очень пестрой.

## Консоль

Для украинизации консоли лучше использовать пакет ruscii от Igor Romanenko. Перекодировка из koi8-u осуществляется на лету. Т.е. ввод и хранение информации происходит в koi8-u, а отображение на экране — в ruscii. Такая странная на первый взгляд техника применяется для того, чтобы символы псевдографики изображались корректно, например в Midnight Commander (mc)<sup>4</sup> или Demos Commander (deco).

Берется пакет console-koi8u.tgz<sup>5</sup> и разворачивается относительно /usr/lib/kbd или другого каталога, где находятся шрифты и раскладки клавиатуры для консоли.

```
tar -zxvf console-koi8u.tgz -C /usr/lib/kbd
```

Теперь следует положить файл keytable<sup>6</sup> в /etc/rc.d/init.d или сверить его с уже имеющимся там. Не забыть установить права доступа:

```
chmod a+x keytable
```

В /etc/rc.d/rc?.d проверить наличие ссылок S??keytable на /etc/rc.d/init.d/keytable. А в /etc/sysconfig поместить файл keyboard. Предполагается, что в компьютере видеоадаптер VGA или лучше. Для EGA в файле /etc/sysconfig/keyboard в строке, содержащей  
SCREENFONT="/usr/lib/kbd/consolefonts/RUSCII\_8x16"  
заменяется 8x16 на 8x14, а для CGA — на 8x8.

Во время очередной загрузки компьютера автоматически будет активизирована поддержка koi8-u. Переключение клавиатуры осуществляется правой клавишей Ctrl для русского языка и правой клавишей Alt — для украинского. Для ручной активизации следует выполнить команду  
/etc/rc.d/init.d/keytable start

Следует заметить, что ввод русских или украинских букв в командной строке (в shell) невозможен без правильной настройки файла ~/.inputrc

Для корректной работы joe его следует запускать с ключом -asis. Для использования less надо установить переменную окружения LESSCHARSET — например, в bash это выглядит так:  
export LESSCHARSET=koi8-r

---

<sup>1</sup> <http://sunsite.unc.edu/LDP/HOWTO/Cyrillic-HOWTO.html>

<sup>2</sup> <http://www.nagual.pp.ru/~ache/koi8.html>

<sup>3</sup> <ftp://ftp.isi.edu/in-notes/rfc2319.txt>

<sup>4</sup> <http://mc.blackdown.org/mc>

<sup>5</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/Linux/console-koi8u.tgz>

<sup>6</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/Linux/keytable>

## Локализация

Локализация нужна для правильной сортировки, вывода денежной единицы, даты и времени в локальном формате. Для работы locales требуется Linux libc 5.4.x с поддержкой locales или glibc 2.x. Дальнейшее описание приводится для Linux libc 5.4.33. locale-ua-1.2.tgz<sup>7</sup> [03.05.98] разворачивается в каталог /usr/share/locale командой

```
tar -zxvf locale-ua-1.2.tgz -C /usr/share/locale
```

**[05.08.98]** В случае с glibc надо использовать locale-ua-glibc-1.2.tgz<sup>8</sup>. Для установки следует произвести аналогичные предыдущим действия:

```
tar -zxvf locale-ua-glibc-1.2.tgz -C /usr/share/locale
```

Этот пакет состоит из uk\_UA и ru\_UA locales. Первая часть в названии означает язык, на котором будет "говорить" система, а вторая — регион, в котором будет использоваться эта локализация. Таким образом uk\_UA содержит правила сортировки украинских букв, названия денежной единицы, дней недели и месяцев на украинском языке, в то время, как в ru\_UA все названия на русском языке.

Для того, чтобы сообщить приложениям, поддерживающим локализацию, о текущей ее установке, надо установить переменную окружения LANG (если устанавливаются отдельные части локализации, то – LC\_COLLATE для сортировки, LC\_CTYPE для классификации символов, LC\_MESSAGES для определения возможных вариантов ответов "да/нет", LC\_MONETARY для описания денежной единицы, LC\_NUMERIC для числовых разделителей, LC\_TIME для описания даты и времени и LC\_ALL для всего вместе):

```
export LANG=uk_UA
```

Проверить текущую локализацию можно командами locale и cal. Первая выдает установки вышеописанных переменных окружения, а вторая выводит календарь на текущий месяц — в случае правильной работы locales названия месяца и дней недели будут на украинском для uk\_UA и на русском для ru\_UA.

Также вышеуказанные locales можно получить из исходных текстов locale-ua-1.2.src.tgz<sup>9</sup> [03.05.98] командами:

```
tar -zxvf locale-ua-1.2.src.tgz -C /usr/share/i18n
```

```
cd /usr/share/i18n/locale
```

```
localedef -ci uk_UA -f KOI8-U uk_UA
```

```
localedef -ci ru_UA -f KOI8-U ru_UA
```

После чего в каталогах /usr/share/locale/uk\_UA и /usr/share/locale/ru\_UA появятся соответственно uk\_UA и ru\_UA locales. Следует отметить, что местоположение файлов локализации можно узнать по команде

```
localedef --help
```

**[05.08.98]** В glibc-версии названия каталогов с исходными файлами локализации в /usr/share/i18n изменились: charmap на charmaps и locale на locales. Поэтому устанавливать надо файл locale-ua-glibc-1.2.tgz также, как описано выше. Кроме измененных названий каталогов в нем отсутствует файл en\_DK, который используется как шаблон в ru\_UA и uk\_UA и который входит в поставку. Также в RedHat 5.1 стали использовать локализацию такие утилиты, как ls и date.

**[03.05.98]** В связи с выходом rfc на koi8-u locales были немного изменены. Также ua\_UA был переименован в uk\_UA.

## X-Window

Украинизация X-Window, как и консоли, состоит из установки экранных шрифтов и переключателя клавиатуры. Все сказанное ниже относится к XFree86, в других версиях некоторые моменты могут отличаться. Font server в X-Window поддерживает как масштабируемые, так и немасштабируемые шрифты. Их в Internet находится достаточное количество, как русских, так и украинских.

<sup>7</sup> [ftp://ftp.cn.ua/pub/unix/koi8-u/Linux/locale-ua-1.2.tgz](http://ftp.cn.ua/pub/unix/koi8-u/Linux/locale-ua-1.2.tgz)

<sup>8</sup> [ftp://ftp.cn.ua/pub/unix/koi8-u/Linux/locale-ua-glibc-1.2.tgz](http://ftp.cn.ua/pub/unix/koi8-u/Linux/locale-ua-glibc-1.2.tgz)

<sup>9</sup> [ftp://ftp.cn.ua/pub/unix/koi8-u/Linux/locale-ua-1.2.src.tgz](http://ftp.cn.ua/pub/unix/koi8-u/Linux/locale-ua-1.2.src.tgz)

Для установки Cronyx шрифтов koi8-r надо взять пакет fonts-2.0.tgz<sup>10</sup> и развернуть его в каталог /usr/X11R6/lib/X11/fonts следующей командой:

```
tar -zxvf fonts-2.0.tgz -C /usr/X11R6/lib/X11/fonts
```

Также эти шрифты можно получить из исходных текстов fonts-2.1.1-src.tgz<sup>11</sup>:

```
tar -zxvf fonts-2.1.1-src.tgz
cd xrus-2.1.1-src
```

Затем поправить Makefile: снять комментарий со строки DIRS = 75dpi 100dpi misc, закомментировать строку DIRS = 75koi8 75dpi 100dpi misc и запустить

```
make X11R6
```

Украинские Gemini koi8-u шрифты устанавливаются из пакета gemini-koi8-u.tgz<sup>12</sup> в каталог /usr/X11R6/lib/X11/fonts

```
tar -zxvf gemini-koi8-u.tgz -C /usr/X11R6/lib/X11/fonts
```

Исходные тексты этих шрифтов находятся в пакете X-koi8u.tgz<sup>13</sup> и устанавливаются следующим образом:

```
mkdir X-koi8u
tar -zxvf X-koi8u.tgz -C X-koi8u
cd X-koi8u
make
make install
```

В файле /etc/X11/XF86Config в секции "Files" исправить параметр FontPath — в начало добавить пути к новым шрифтам:

Section "Files"

```
RgbPath      "/usr/X11R6/lib/X11/rgb"
FontPath     "/usr/X11R6/lib/X11/fonts/ukr"
FontPath     "/usr/X11R6/lib/X11/fonts/cyrillic/misc"
FontPath     "/usr/X11R6/lib/X11/fonts/cyrillic/75dpi"
FontPath     "/usr/X11R6/lib/X11/fonts/cyrillic/100dpi"
FontPath     "/usr/X11R6/lib/X11/fonts/Type1"
FontPath     "/usr/X11R6/lib/X11/fonts/Speedo"
FontPath     "/usr/X11R6/lib/X11/fonts/misc"
FontPath     "/usr/X11R6/lib/X11/fonts/75dpi"
FontPath     "/usr/X11R6/lib/X11/fonts/100dpi"
```

EndSection

В каталоге /usr/X11R6/lib/X11/fonts/ukr находятся koi8-u gemini шрифты, а в каталогах /usr/X11R6/lib/X11/fonts/cyrillic/\* — koi8-r cronux шрифты. Следует обратить внимание на порядок указания путей: так во время использования какой-нибудь программой шрифта без указания его точных параметров (foundry, registry, encoding и т.д.) шрифт будет искажаться сначала в каталоге ukr, затем в cyrillic, а затем в остальных каталогах.

Для переключения клавиатуры с одной раскладки на другую следует установить пакет xruskb-1.5.0.tar.gz<sup>14</sup> от Alexander V. Lukyanov. Для его компиляции нужны библиотеки Motif, Lesstif или Xaw. В случае отсутствия этих библиотек xrus все-равно будет работать, но текущее состояние раскладки клавиатуры на экране отображаться не будет.

```
tar -zxvf xruskb-1.5.0.tar.gz
cd xruskb-1.5.0
./configure
make
make install
```

<sup>10</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/fonts-2.0.tgz>

<sup>11</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/fonts-2.1.1-src.tgz>

<sup>12</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/gemini-koi8-u.tgz>

<sup>13</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/X-koi8u.tgz>

<sup>14</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/xrus/xruskb-1.5.0.tar.gz>



После этого в каталог `/usr/local/xrus` (или в тот, в который проинсталлировался `xrus`) разворачивается `jcukn-ukr.tgz`<sup>15</sup>, в `/usr/X11R6/lib/X11/app-defaults` переписывается файл `Xrus`<sup>16</sup> и в файле `/etc/X11/xinit/xinitrc` перед запуском `vwm` (Virtual Window Manager) добавляется команда запуска `xrus`.

Уже откомпилированные версии `xrus 1.4.2` с динамической и статической поддержкой Motif (`xrus.dynamic-Motif.tgz`<sup>17</sup> и `xrus.static-Motif.tgz`<sup>18</sup> соответственно) разворачиваются в каталог `/usr/local/xrus` и настраиваются так-же, как описано выше для пакета с исходными текстами.

## Netscape

К сожалению, ни Netscape Navigator, ни Netscape Communicator ничего не знают про `koi8-u`, т.к. `rfs` на `koi8-u` еще не принят. Для просмотра WWW и чтения писем/новостей на украинском языке можно настроить кодировку User-Defined — указать `koi8-u` шрифты в Edit->Preferences->Fonts. Но писать в этой кодировке не следует, так как не все смогут это прочесть.

Все остальное, вроде списка новостей и писем, списка папок и конференций, всплывающих подсказок и т.д., можно украинизировать путем установки файла `.Xdefaults`<sup>19</sup> в домашний каталог или добавлением его к уже имеющемуся.

## Apache

Для корректного представления web-страниц в различных кодировках надо установить Russian Apache `apache_1.2.6rusPL22.2.tar.gz`<sup>20</sup> и в его конфигурационный каталог (обычно `/etc/httpd/conf`) развернуть украинские таблицы перекодировки `apache-ukr-tables.tgz`<sup>21</sup>:

```
tar -zxvf apache-ukr-tables.tgz -C /etc/httpd/conf
```

## Wabi

Wabi 2.0 представляет собой эмулятор MS Windows 3.x. Он использует Windows ttf-шрифты, которые уже содержат украинские буквы. А для переключения клавиатуры можно использовать пакет `xruskb` с украинскими раскладками `jcukn-ukr`, которые описываются в разделе X-Window и содержат русскую и украинскую раскладки в кодировке `cp1251`.

## KDE

Для переключения клавиатуры в KDE (K Desktop Environment) можно использовать пакет `xruskb`, описанный в разделе X-Window, но существует специально созданная для этого Александром Будником программа `kikbd-0.1.6.tgz`<sup>22</sup>. После инсталляции, в каталог `/usr/local/kde/share/apps/kikbd` разворачивается пакет украинских раскладок `koi8-u` и `cp1251` `kikbd-ukr.tgz`<sup>23</sup>.

**[24.02.98]** Новая версия 0.1.7 уже содержит украинские раскладки клавиатуры и другие полезные новшества.

**[21.04.98]** Последняя версия — `kikbd-0.2.1.tgz`<sup>24</sup>

## PostgreSQL

В данном разделе описывается не украинизация, а доработка PostgreSQL для работы с разными кириллическими кодировками, в том числе и украинскими, такими как `koi8-u`, `cp1251` и т.д. Дело в том, что клиент SQL может работать на любой платформе, например в Windows через

<sup>15</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/xrus/jcukn-ukr.tgz>

<sup>16</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/xrus/Xrus>

<sup>17</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/xrus/xrus.dynamic-Motif.tgz>

<sup>18</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/xrus/xrus.static-Motif.tgz>

<sup>19</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/.Xdefaults>

<sup>20</sup> <ftp://apache.lexa.ru/pub/apache-rus/>

<sup>21</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/apache-ukr-tables.tgz>

<sup>22</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/kikbd-0.1.6.tgz>

<sup>23</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/X11/kikbd-ukr.tgz>

<sup>24</sup> <ftp://ftp.kde.org/pub/kde/unstable/apps/utls/kikbd-0.2.1.tgz>

ODBC и использовать при этом кодировку cp1251, в то время, как сервер обычно использует koi8 кодировку. Данный патч призван решить эту проблему путем перекодировки информации из одной кодовой страницы в другую в зависимости от адреса клиента. `pgsql-charsets-6.3-4.patch.gz` устанавливается на исходный код PostgreSQL 6.3 beta (snapshot от 18.02.98 или позднее) в каталоге `/usr/src/pgsql` командами:

```
gzip -d pgsql-charsets-6.3-4.patch.gz
mv pgsql-charsets-6.3-4.patch /usr/src/pgsql/src
cd /usr/src/pgsql/src
patch -p0 <pgsql-charsets-6.3-4.patch
autoconf
./configure --enable-recode
gmake all
gmake install
```

**[10.03.98]** Релиз PostgreSQL 6.3 содержит вышеописанный патч по перекодировке! К большому сожалению и по непонятной причине, информация в файлах каталога `src/data` оказалась "продублирована". Это не сказывается на работе перекодировщика, но факт остается фактом...

Как и в случае с патчем, для включения перекодировщика следует запустить:

```
./configure --enable-recode
gmake all
gmake install
```

**[21.04.98]** В версии 6.3.1 вышеописанный недочет был исправлен. Последняя версия — PostgreSQL 6.3.2

## PostScript

Работы по руссификации/украинизации PostScript в Linux, находятся по адресу <ftp://ftp.cn.ua/pub/unix/koi8-u/fonts>. В этом разделе архива собраны русско-украинские шрифты. PostScript шрифты находятся в архиве `Type1.tgz`<sup>25</sup>. Скрипты, которые необходимы для того, чтобы заставить Linux печатать по русски на принтер находятся в каталоге `utils`. Рекомендую прочитать файлы `README` в соответствующих каталогах, для более детального объяснения. При возникновении каких-либо вопросов касающихся работе с PS-Type1, пожалуйста пишите по адресу [hunter@cn.ua](mailto:hunter@cn.ua)

## Ссылки

KOI8-U — Home Page (<http://www.net.ua/KOI8-U>)

KOI8-U — Ukrainian Character Set (<http://cad.ntu-kpi.kiev.ua/multiling/koi8-u>)

Unix and Linux localization (<http://www.linux.org.ua/>)

## Курс молодого бойца. Основы управления доступом к файлам

Автор статьи Lomash Dmitry ([lomash@mtuci-ncb.donpac.ru](mailto:lomash@mtuci-ncb.donpac.ru))

*Операционная система Linux — это многопользовательская система, которая дает огромные возможности манипулирования доступом к данным для каждого пользователя отдельно. Это позволяет гибко регулировать отношения между пользователями, объединяя их в группы, что позволит защитить данные одного пользователя от нежелательного вмешательства других.*

Бессмысленно считать, что файловая система это не самая важная часть операционной системы, поскольку все данные пользователей хранятся именно в файлах.

В UNIX-подобных системах файлы также обеспечивают доступ к периферийным устройствам, дисковым накопителям, принтерам и т.п.

<sup>25</sup> <ftp://ftp.cn.ua/pub/unix/koi8-u/fonts/Type1.tgz>



## UID, GID

Каждый пользователь в системе имеет свой уникальный идентификационный номер (user-ID, или UID). Также пользователи могут объединяться в группы, которые в свою очередь имеют group-ID, или GID. Чтобы узнать свой UID и GID, т.е. уникальный номер пользователя и номер группы, к которой Вы принадлежите, необходимо ввести команду id:

```
[dmitry@localhost dmitry]$id
uid=502 (dmitry) gid=503(users) groups=503(users)
```

## Права доступа к файлам

В свою очередь файлы имеют двух владельцев: пользователя (user owner) и группу пользователей (group owner). Для каждого файла есть индивидуальные права доступа, которые разбиты на три группы:

1. Доступ для пользователя-владельца файла (owner).
2. Доступ для группы-владельца файла (group).
3. Доступ для остальных пользователей (others).

Для каждой категории устанавливаются три вида доступа: (x) — право на запуск файла, (r) — право на чтение файла, (w) — право на изменение (редактирование) файла.

Для того, чтобы увидеть права доступа к файлам необходимо ввести команду ls с ключом -l:

```
[dmitry@localhost dmitry]$ls -l /home/file.tmp
-rwxr-xr-- 1 dmitry users 33 Dec 1 00:38 file.tmp
```

Для данного примера мы видим, что владелец имеет права на чтение, запись, и выполнение (первые три буквы rwx), группа пользователей может лишь читать и выполнять этот файл (следующие три r-x), ну а остальные пользователи могут только читать данный файл (последние символы r--).

## Изменение прав доступа

Права пользователя могут быть изменены только владельцем файла или пользователем с правами администратора системы. Для изменения прав используется команда `chmod [ u | g | o | a ] [ + | - | = ] [ r | w | x ] name1 [ name2 ... ]`.

В качестве аргументов команда принимает указание классов доступа ('u' — владелец-пользователь, 'g' — владелец-группа, 'o' — остальные пользователи, 'a' — все вышеперечисленные группы вместе), права доступа ('r' — чтение, 'w' — запись, 'x' — выполнение) и операцию, которую необходимо произвести ('+' — добавить, '-' — убрать, '=' — присвоить).

Таким образом, чтобы разрешить выполнение файла `prog.pl` всем пользователем необходимо выполнить команду:

```
[dmitry@localhost dmitry]$ chmod a+x prog.pl
```

Далее, чтобы оставить права записи только для владельца файла необходимо выполнить:

```
[dmitry@localhost dmitry]$ chmod go-w prog.pl
```

Рассмотрим еще несколько примеров:

\$ chmod go=w prog.pl	установить право на запись для всех пользователей кроме владельца
\$ chmod a+x prog.pl	предоставить право на запись для всех пользователей
\$ chmod g+x-w prog.pl	Добавить для группы право на выполнения файла, но снять право на запись

Права доступа можно представить в виде битовой строки, в которой каждые 3 бита определяют права доступа для соответствующей категории пользователей, как представлено в таблице:

gwx	gwx	gwx
421	421	421
user	group	others
владелец	группа	остальные

Таким образом, для команды `chmod 666 prog.pl` имеем:

```
[dmitry@localhost dmitry]$ chmod 666 prog.pl  
[dmitry@localhost dmitry]$ ls -l prog.pl  
-rw-rw-rw- 1 dmitry users 78 Nov 20 prog.pl
```

Команда `chmod 644 somefile` устанавливает "обычные" права доступа, т.е. владелец может читать и записывать в файл, а все остальные пользователи — только читать.

### Особенности прав доступа для каталогов

Права доступа для каталогов не столь очевидны. Это в первую очередь связано с тем, что система трактует операции чтения и записи для каталогов отлично от остальных файлов. Право чтения каталога позволяет Вам получить имена (и только имена) файлов, находящихся в данном каталоге. Чтобы получить дополнительную информацию о файлах каталога (например, подробный листинг команды `ls -l`), системы придется "заглянуть" в метаданные файлов, что требует права на выполнения для каталога. Право на выполнение также потребуется для каталога, в который Вы захотите перейти (т.е. сделать его текущим) с помощью команды `cd`.

### T-бит, SUID и SGID

Наиболее внимательные пользователи быстро замечают, что помимо стандартных "rwx" значений существуют еще и буквы "s" и "t". В действительности, битовая маска прав доступа к файлам содержит 4 группы по 3 бита в каждой. Таким образом, команда `chmod 755` это всего лишь краткая запись полной формы команды: `chmod 0755`.

t-бит обычно используется с каталогами.

Обычно, когда t-бит для каталога не установлен, файл в данном каталоге может удалить любой пользователь, имеющий доступ на запись к данному файлу. Устанавливая t-бит на каталог мы меняем это правило таким образом, что удалить файл из каталога может только владелец этого каталога или файла.

Установить t-бит можно при помощи команд `chmod a+tw somefile` или `chmod 1777 somefile`.

Атрибуты SUID и SGID позволяют изменить права пользователя при запуске на выполнения файла, имеющего эти атрибуты.

Запускаемая программа получает права доступа к системным ресурсам на основе прав доступа пользователя, запустившего программу. Установка же флагов SUID и SGID изменяет это правило таким образом, что назначает права доступа к системным ресурсам исходя из прав доступа владельца файла. Т.е. запущенный исполняемый файл, которым владеет суперпользователь, получает права доступа к системным ресурсам на уровне суперпользователя (фактически неограниченные). При этом установка SUID приведет к наследованию прав владельца-пользователя файла, а установка SGID - владельца группы.

В завершении хочется отметить, что пользоваться такими мощными атрибутами как SUID и SGID нужно с крайней осторожностью, особенно подвергать пристальному вниманию программы и скрипты, владельцем которых является root (суперпользователь), т.к. это потенциальная угроза безопасности системы.

## Изучаем файловую систему proc

Sandeep Grover, [sandeep@Magma-DA.com](mailto:sandeep@Magma-DA.com)

Перевод Pukhlyakov Kirill, [kirill@linuxfocus.org](mailto:kirill@linuxfocus.org)

Ядро Linux предоставляет механизм доступа к своим внутренним структурам и позволяет изменять установки ядра во время работы ОС посредством системы /proc. Мы рассмотрим в заметке эту систему касательно архитектуры Intel x86, но основные принципы применимы к ОС Linux на любой платформе.

### Виртуальная файловая система /proc

Файловая система /proc является механизмом для ядра и его модулей, позволяющим посылать информацию процессам (отсюда и название /proc). С помощью этой виртуальной файловой системы Вы можете работать с внутренними структурами ядра, получать полезную

информацию о процессах и изменять установки (меняя параметры ядра) на лету. Файловая система `/proc` располагается в памяти в отличие от других файловых систем, которые располагаются на диске. Если Вы посмотрите на файл `/proc/mounts` (он выводит список всех подмонтированных файловых систем подобно команде "mount") то увидите что-нибудь типа :

```
grep proc /proc/mounts
/proc /proc proc rw 0 0
```

`/proc` контролируется ядром. Из-за того, что она предоставляет информацию контролируемую ядром, логично, что она располагается в памяти, контролируемой также ядром. Команда "ls -l" покажет, что большинство файлов в этой системе имеют нулевую длину, но посмотрев любой файл Вы получите достаточно информации. Как это может быть? Все просто — файловая система `/proc` как любая другая файловая система регистрируется на уровне VFS (Virtual File System layer). Поэтому при запросе файлов/каталогов, файловая система `/proc` создает эти файлы/каталоги на основании информации, содержащейся в ядре.

### Монтирование файловой системы /proc

Если `/proc` еще не подмонтирована - Вы можете сделать это следующим образом:

```
mount -t proc proc /proc
```

Более детальную информацию Вы получите, прочитав `man mount`.

### Просмотр файлов /proc

С помощью файлов в `/proc` Вы можете получить информацию о состоянии ядра, процессов, параметрах компьютера и т.д. Большинство файлов в `/proc` содержат самую свежую информацию о системном оборудовании. Несмотря на то, что эти файлы виртуальные — их можно просмотреть любым текстовым редактором или с помощью команд "more", "less" или "cat". При попытке открытия виртуального файла текстовым редактором — этот файл создается на лету на основе информации, содержащейся в ядре. Приведу здесь некоторые интересные цифры о моей системе:

```
$ ls -l /proc/cpuinfo -r--r--r-- 1 root root 0 Dec 25 11:01 /proc/cpuinfo
$ file /proc/cpuinfo
/proc/cpuinfo: empty

$ cat /proc/cpuinfo

processor           : 0
vendor_id          : GenuineIntel
cpu family         : 6
model              : 8
model name         : Pentium III (Coppermine)
stepping           : 6
cpu MHz            : 1000.119
cache size         : 256 KB
fdiv_bug           : no
hlt_bug            : no
sep_bug            : no
f00f_bug           : no
coma_bug           : no
fpu                : yes
fpu_exception      : yes
cpuid level        : 2
wp                 : yes
flags               : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca Å
                    cmov pat pse36 mmx fxsr xmm
bogomips           : 1998.85
processor           : 3
vendor_id          : GenuineIntel
cpu family         : 6
model              : 8
model name         : Pentium III (Coppermine)
stepping           : 6
```

```

cpu MHz      : 1000.119
cache size   : 256 KB
fdiv_bug     : no
hlt_bug      : no
sep_bug      : no
f00f_bug     : no
coma_bug     : no
fpu          : yes
fpu_exception : yes
cpuid level   : 2
wp           : yes
flags        : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca Å
               cmov pat pse36 mmx fxsr xmm
bogomips     : 1992.29

```

Это двухпроцессорная система. В некоторых файлах файловой системы /proc информация закодирована, однако некоторые утилиты приспособлены к этому — они выводят ее в виде понятном человеку. К этим утилитам относятся: "top", "ps", "apm" и т.д.

### Полезная информация о системе и ядре

Перечислим некоторые важные файлы:

- Ø /proc/cpuinfo — информация о процессоре ( модель, семейство, размер кэша и т.д.)
- Ø /proc/meminfo — информация о RAM, размере свопа и т.д.
- Ø /proc/mounts — список подмонтированных файловых систем.
- Ø /proc/devices — список устройств.
- Ø /proc/filesystems — поддерживаемые файловые системы.
- Ø /proc/modules — список загружаемых модулей.
- Ø /proc/version — версия ядра.
- Ø /proc/cmdline — список параметров, передаваемых ядру при загрузке.

На самом деле файлов в каталоге /proc намного больше. Вы можете детально просмотреть все файлы в /proc командами "more" и read [1]. Советую Вам использовать "more" вместо "cat" потому, что некоторые файлы ( например kcore ) могут быть очень большого размера.

### Информация о процессах

Файловая система /proc служит также источником информации о выполняющихся процессах. Если Вы внимательно посмотрите внутрь каталога /proc, то увидите каталоги, названия которых состоят из цифр — это и есть информация о процессах — название каталога отражает идентификатор процесса (PID). Внутри этих каталогов находятся файлы, содержащие важную информацию о процессах — состояние, окружение и т.д. Давайте посмотрим на какой-нибудь процесс:

```

$ ps -aef | grep mozilla
root 32558 32425 8 22:53 pts/1 00:01:23 /usr/bin/mozilla

```

Выполненная нами команда показывает, что в системе присутствует процесс mozilla, его PID 32558. Соответственно в каталоге /proc должен быть каталог с названием 32558.

```

$ ls -l /proc/32558
total 0
-r--r--r-- 1 root root 0 Dec 25 22:59 cmdline
-r--r--r-- 1 root root 0 Dec 25 22:59 cpu
lrwxrwxrwx 1 root root 0 Dec 25 22:59 cwd -> /proc/
-r----- 1 root root 0 Dec 25 22:59 environ
lrwxrwxrwx 1 root root 0 Dec 25 22:59 exe -> /usr/bin/mozilla*
dr-x----- 2 root root 0 Dec 25 22:59 fd/
-r--r--r-- 1 root root 0 Dec 25 22:59 maps
-rw----- 1 root root 0 Dec 25 22:59 mem
-r--r--r-- 1 root root 0 Dec 25 22:59 mounts
lrwxrwxrwx 1 root root 0 Dec 25 22:59 root -> //
-r--r--r-- 1 root root 0 Dec 25 22:59 stat

```

-r--r--r--	1	root	root	0	Dec 25 22:59	statm
-r--r--r--	1	root	root	0	Dec 25 22:59	status

В файле "cmdline" содержится информация, которой был запущен процесс. В файле "environ" находятся переменные окружения этого процесса, а также идентификатор пользователя (UID) и группы (GID) запустившего процесс, идентификатор родительского процесса (PPID) и текущее состояние процесса (например "Sleeping" или "Running"). Также в подобных каталогах находятся символические ссылки: "cwd" — ссылка к текущему рабочему каталогу процесса, "exe" — исполняемый файл процесса, "root" — ссылка к каталогу, который считается процессом корневым (обычно это "/"). В каталоге "fd" находятся ссылки к используемым дескрипторам файлов процессом. "cpu" доступен только на SMP ядрах Linux.

/proc/self интересный подкаталог — с его помощью приложение найдет информацию о себе. На самом деле /proc/self является символической ссылкой на каталог процесса обращающегося к /proc.

### Работа с ядром посредством /proc

Большинство из рассмотренных нами файлов из каталога /proc доступны только на чтение. Тем не менее файловая система /proc предоставляет файлы для взаимодействия с ядром доступные на запись. Помещение информации в такие файлы изменяет состояние ядра и поэтому это надо делать очень аккуратно. Эти файлы находятся в каталоге /proc/sys.

В каталоге /proc/sys/kernel находится информация общего плана для ядра. Соответственно в /proc/sys/kernel/{domainname, hostname} находится информация о доменном имени и host имени, которую Вы можете изменить.

```
$ hostname
machinename.domainname.com

$ cat /proc/sys/kernel/domainname
domainname.com

$ cat /proc/sys/kernel/hostname
machinename

$ echo "new-machinename" > /proc/sys/kernel/hostname

$ hostname
new-machinename.domainname.com
```

Таким образом изменив файл в файловой системе /proc мы можем поменять hostname. Много данных для настройки находится в /proc/sys/kernel/. Повторюсь, невозможно в заметке рассмотреть все файлы, думаю Вам самим будет интересно повнимательней посмотреть на эти каталоги.

Еще один интересный каталог /proc/sys/net. В нем Вы можете произвести сетевые настройки, например спрятать Ваш компьютер в сети.

```
$ echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

Запретив ответы на icmp\_echoes мы спрятали компьютер. Наша машина не будет отвечать на команды "ping" других компьютеров.

```
$ ping machinename.domainname.com
no answer from machinename.domainname.com
```

Вернуть прежнее состояние можно следующим образом

```
$ echo 0 > /proc/sys/net/ipv4/icmp_echo_ignore_all
```

В каталоге /proc/sys еще много возможностей для настройки системы. Более подробная информация находится в [1] и [2].

## Вывод

Файловая система /proc обеспечивает файловый интерфейс к внутренним структурам Linux. С ее помощью можно узнать полезную информацию и настроить систему. Знание этой файловой системы и умение применить эти знания — большой плюс в работе с ОС Linux.

## Список литературы

[1] Документация по файловой системе ОС Linux находится здесь:

/usr/src/linux/Documentation/filesystems/proc.txt

[2] Руководство пользователя RedHat: Файловая система /proc:

<http://www.redhat.com/docs/manuals/linux/RHL-7.3-Manual/ref-guide/ch-proc.html>

## Блины со сметаной

Сергей А.ЯРЕМЧУК, [grinder@ua.fm](mailto:grinder@ua.fm)

*Я уже не раз подчеркивал в своих статьях: свой дистрибутив Linux может создать кто угодно, было бы желание. Наверное, и ребятам из Запорожья не захотелось сидеть сложа руки и ждать, когда же им подкинут дистрибутивчик, полностью удовлетворяющий их вкусам и запросам, а взяли и «слепили из того что было» свой запорожский Linux. И что вдвойне отрадно, ведь с тех пор, как разработчики украинского BlackCat из Донецка переехали в Россию и назвали себя ASPLinux, у нас официально не стало «своего» Linux'a.*

*Linux - это не птица, а сорт пива!  
(Сайт Запорожской Linux User Group)*

В принципе, отсутствие отечественного дистрибутива Linux не то чтобы совсем уж смертельно, ведь в российских дистрибутивах учтены интересы наших пользователей, более того, предпринимаются попытки локализации (украинизации) не только графических сред, но и других программ, с которыми будет работать «користувач». Но наличие своей операционной системы позволяет решить множество проблем, в том числе и государственных, но я остановлюсь лишь на близких к пользователю. Первая - покупка: тот, кто пытался купить дистрибутив в России, может столкнуться с некоторыми неудобствами, в том числе и в оплате; может быть, для крупных городов вроде Киева это не так существенно, а в других придется немножечко побегать. Вторая - доставка: что ни говори, а Украина уже за граница, и заказы, скорее всего, обрабатываются во вторую очередь; у меня со дня заказа до получения ни когда не получалось менее 45 дней, а то и больше. Ну кто, скажите, захочет ждать так долго? И третье – в последнее время стало хорошим тоном добавлять в дистрибутивы программные продукты, выпущенные местными программистами, в том числе и демоверсии, в рекламных целях. Это хорошо, но, например, их бухгалтерские программы в наших условиях бесполезны, поэтому в большинстве случаев приходится искать замену. А своим помогать? Во, блин...

Нет, это я не ругаюсь – именно так называется дистрибутив Linux, который можно найти на сайте запорожской **Linux User Group (LUG)** <http://linux.zp.ua>. Название **Blin**, как я понял, произошло от **BCS Linux** (Business Computer Service) - фирмы или организации, поддерживающей сайт. Единственное, что огорчает при походе на сайт, это то, что их сервер тащится как губа по стекловате. Ждать, пока страница (~50 Кб) загрузится, целых десять минут, а то и более – это слишком. При том, что счетчик показывал всего четыре посетителя. А ведь после прочтения этой статьи их наверняка будет больше... В общем, если хватит терпения, на сайте вы найдете информацию как о самом дистрибутиве, а также о том, где и как можно купить или скачать Blin и как испечь самые настоящие блины.



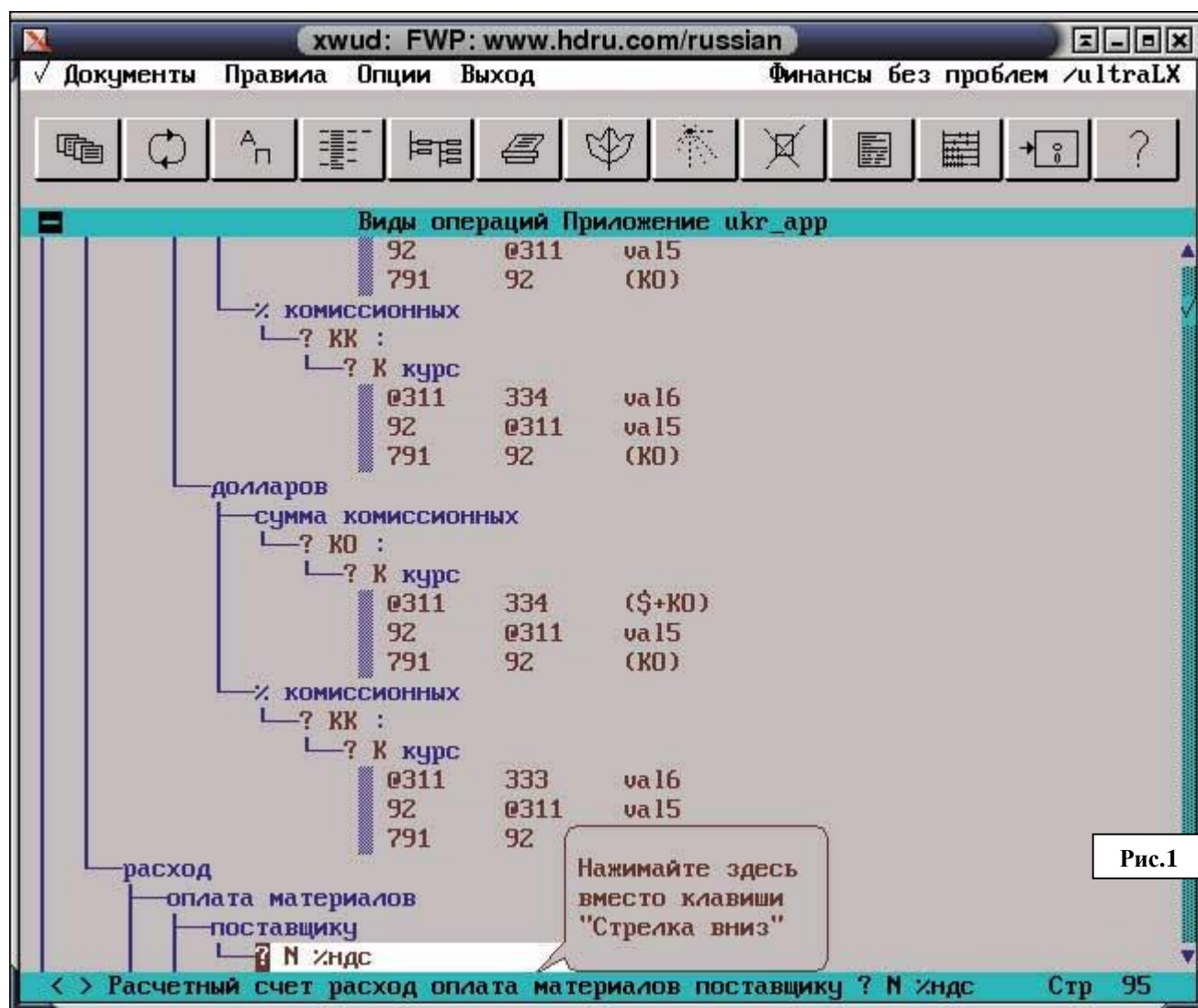
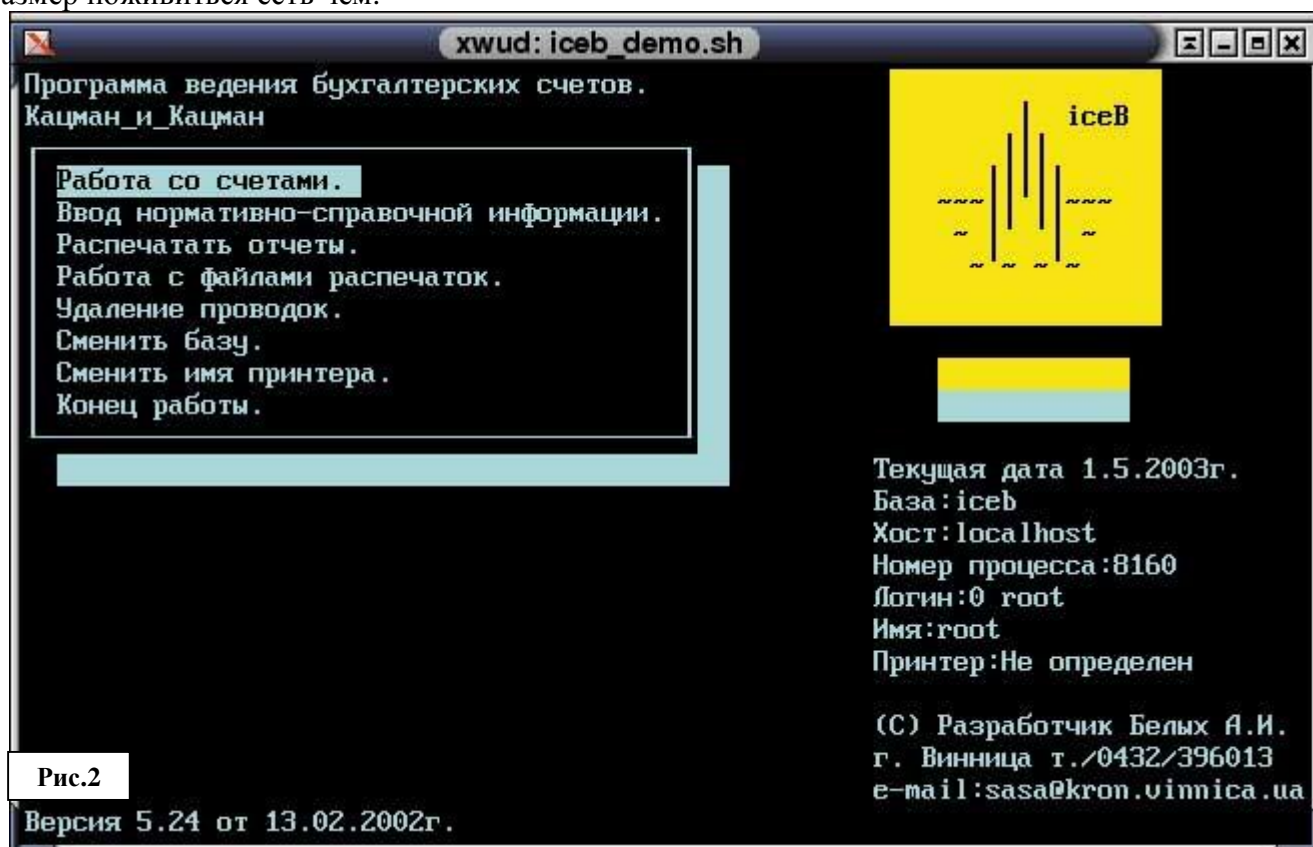


Рис.1

Итак, что же имеем в данном случае. Blin относится к популярным в последнее время Live-CD дистрибутивам, т.е. для своей работы не требует установки на жесткий диск. При загрузке система сама находит все разделы жесткого диска и монтирует их в режиме только для чтения. В дистрибутиве используется файловая система устройств `devfs`, но автоматическое создание ссылок дает возможность пользоваться более удобными старыми обозначениями. Большинство приложений, используемых в blin (кроме тех, которые по определению не имеет смысла заставлять работать под слабой конфигурацией, вроде гномых утилит или Mozilla), собраны под i386-процессор, что позволяет ставить его на достаточно старые компьютеры. В своем составе blin имеет ядро **2.4.19**, **glibc-2.2.5**, **XFree86 4.2**, в качестве оконного менеджера использован **Windowmaker-0.80.1**, для прослушивания музыки предоставляется **XMMS**, **mpg123** и полный комплект для работы с Ogg Vorbis; фильм можно просмотреть при помощи **Mplayer**. Есть средства для просмотра графических файлов – **GQView** и **Eye Of Gnome (eog)**. Для работы с разделами жесткого диска заготовлен полный комплект соответствующих утилит — **e2fsprog**, **raiserfsprogs**, а также **parted**, **cfdisrk**, **sfdisk** и **mttools** для работы с DOS-разделами. На сайте сказано, что Blin в первую очередь ориентирован на подготовленного пользователя (хотя это заявление не должно отпугнуть бывалых новичков), для решения системных задач и работы в сети. Действительно, для этих целей в комплект дистрибутива включен целый набор утилит: `ssh`, `telnet`, `citrix`, `rdesktop`, `dial-up`, `smbclient`, `nmap`, `tcpdump`, а также web-браузеры `galeon`, `mozilla`, `links` и утилиты для загрузки файлов – `wget` и графический вариант `WebDownloader for X`. Причем, чтобы каждый раз при новой загрузке системы не настраивать сеть заново, можно поручить это дело скриптам, позволяющим сохранить настройки на дискете и при необходимости загрузить их оттуда. Тем, у кого `callback`, будет полезен одноименный скрипт, позволяющий реализовать данную возможность, аналогично для автодозвона к провайдеру предназначен скрипт `redialer`. В

данный момент blin выпускается в малоформатном исполнении и занимает в объеме 180 Мб; судя по информации на сайте, это сделано специально для того, чтобы иметь возможность записать его на любые 3-х дюймовые CD-R/RW-носители объемом 183 Мб, 193 Мб и 210 Мб (чем я не преминул воспользоваться). В принципе, диск с blin'ом можно использовать в качестве спасательного, как для Linux-систем, так и вообще для работы с жестким диском. И конечно же, blin будет незаменим при настройке сети, по крайней мере, мне уже пригодился. И места много не занимает, так что его можно таскать с собой. Как говорится, «Linux всегда живой, Linux всегда со мной». Дополнительно blin позволит ознакомиться с отечественными бухгалтерскими программами под Linux. В демонстрационном режиме доступны ultraLX «Финансы без проблем» (<http://hdru.com/russian>, Рис. 1) и открытая торгово-бухгалтерская система iceB (<http://www.iceb.vinnitsa.com>, Рис. 2). Текст можно набить в простых редакторах типа gedit, xedit или mcedit (встроенный редактор Midnight Commander). Также имеется ICQ-клиент, samba-браузер LinNeighborhood, почтовый клиент mutt, netconfig (для того чтобы долго не возиться с настройкой локальной сети), планировщики gnomecal и GTimeTracker (gtt), для просмотра файлов в формате .pdf – AcrobatReader 5.0 и xpdf. И напоследок, для досуга 10 игр вроде xbill, gnomine, Gtkballs, gnometrис и пр. Всего доступно свыше 150 пакетов, так что несмотря на кажущийся маленький размер пожить есть чем.



После инициализации системы, в начале которой можно выбрать несколько режимов запуска, но в простейшем случае достаточно нажать Enter, появляется окно рабочего стола Рис. 3. Дополнительно, очевидно для тех, кого может испугать непривычным видом WindowMaker, запускается панель от оконного менеджера Gnome. Интересно, что система при старте выдала сообщение о том, что нашла ТВ-тюнер, но вот программ для работы с ним в комплекте нет, а пока я разобрался, что к чему, тюнер пора уже было отдавать. Текущую клавиатурную раскладку отображает автоматически запускающийся при старте апплет wmlang в виде флага страны. При этом по CapsLock переключается только rus/eng, а чтобы переключиться в украинский, необходимо щелкнуть пару раз по апплету, пока не появится наша «жовто-синя фана», или удерживать правый Alt при выбранной русской раскладке. При нормальной загрузке оперативной памяти у меня было занято не более 32 Мб, если кому-то не будет этого хватать, то можно подручными средствами (или с помощью PartitionMagic) создать своп-раздел (файл подкачки) и примонтировать его после

запуска (swapon). По умолчанию все найденные разделы монтируются в режиме «только для чтения» (всем этим управляет скрипт /usr/sbin/mountnfs), при необходимости можно перемонтировать в режим «чтение-запись». Еще разработчики рекомендуют для повседневного использования утилиту MtoolsFM (Рис. 4), фронт-энд к пакету mtools, позволяющую скидывать информацию на дискету, с дополнительной возможностью архивации и разархивации данных.



Рис.3



Рис.4

После того как наиграетесь с blin'ом, его можно установить на жесткий диск с помощью скрипта (как я понял, собственноручно приготовленного) **install-blin-to-hdd**. При этом скрипт три раза (для недоверчивых) выдаст предупреждение о том, что все данные на диске (первом в системе, т.е. hda) будут уничтожены (для установки нужен чистый диск; немного подработав скрипт, можно уничтожить данные на другом диске :-)), и для подтверждения попросит ввести длинную фразу: *Yes, I want to format my HDD and to lost all previous data*, иначе скрипт закончит свою работу. Если есть желание попробовать, то скопируйте ее при помощи мышки и вставьте по



щелчку средней кнопки в нужное место. Но в моем случае скрипт завершился с ошибкой. Для того чтобы все таки установить blin на жесткий диск, скопируйте скрипт себе в виртуальный домашний каталог (реально он находится на CD-ROM, и исправить его там не получится) и откройте в текстовом редакторе:

```
# cd
# cp /usr/sbin/install-blin-to-hdd .
# gedit ./install-blin-to-hdd
```

Теперь исправьте в двух местах имя файла ядра с mlinuz на vmlinuz:

```
# copy kernel...
cp /mnt/cdrom/isolinux/vmlinuz $mountpoint || die
```

И в строках, где создается файл /etc/lilo.conf, предназначенный для установки параметров загрузчика:

```
cat >$mountpoint/etc/lilo.conf <<EOF || die
boot = /dev/$disk
delay = 50
append = "4"
image = /vmlinuz # и вот тут
root = /dev/${disk}1
label = startup
read-only
EOF
```

К слову сказать, файл /etc/lilo.conf при использовании еще какой-либо операционной системы придется немного опосля подправить, иначе - linux forever.

Все. После этого делаем его исполняемым (chmod +x ./install-blin-to-hdd) и запускаем ./install-blin-to-hdd. После этого в первом гигабайте будут созданы четыре раздела (первичный и три логических). Корневой – 800 Мб, 128 Мб отдано под swap и два раздела по 100 Мб будут отданы под /home и /var (Рис. 5).



Рис.5

Итого 1 Гб, остальное используйте по вашему усмотрению. И все эти разделы будут автоматически прописаны в /etc/fstab, и в конце всего будет установлен LILO. После окончания работы скрипта я перезагрузился, изменил настройки BIOS и в дальнейшем работал с жесткого диска как ни в чем не бывало, без лишних телодвижений. Ну, и раз уже коснулись разбития диска, давайте посмотрим, как все это выглядит в скрипте:

```

disk=hda #эту строчку достаточно заменить, и тогда скрипт будет кромсать другой
#диск, но загрузчик в таком случае придется подправить
mountpoint=/mnt/${disk}1 # сюда смонтированы все разделы на жестком диске
for mnt in /mnt/${disk}*
do
    if [ -e "$mnt" ]
then
umount $mnt || die # теперь они по очереди размонтируются
fi
done
#Building a disklabel(valid DOS partition table).
/bin/dd if=/boot/disklabel_dos of=/dev/$disk || die

```

В этой строке формируется правильная таблица DOS-раздела – файл `disklabel_dos` размером как раз 512 байт, т.е. точно совпадает с размером MBR, куда она и заносится.

И далее при помощи `sfdisk` разбивается диск. Честно говоря, не знал, что `sfdisk` может это проделать так лихо, наверное, весь секрет именно в том, что таблица MBR создается вручную, хотя удобнее все-таки было бы, как мне кажется, воспользоваться `parted`:

```

sfdisk -uM /dev/$disk <<EOF || die
0,800,,*
,329,X
,,0
,,0
,128,S
,100
,,L
EOF

```

Теперь можно заняться подгонкой системы под свои нужды и вкусы. Первоначально советую закомментировать в файле `/etc/rc.M` строчки, из которых вызываются скрипты `/usr/sbin/mounthdd` и `/usr/sbin/blinxsetup`. Первый будет по-прежнему пытаться (в большинстве своем безуспешно, ведь все уже и так смонтировано) найти все оставшиеся разделы на жестком диске, а второй каждый раз при запуске будет тестировать и перенастраивать X-Window и, соответственно, в итоге переписывать заново файл `XF86Config` (в моем случае генерирующийся из шаблонного `/var/etc/XF86Config.tmpl`). После внесения корректив можно спокойно сменить раскладку (я, по крайней мере, всегда использую вариант `winkeys`) и добавить немножечко герц в частоту развертки. Теперь `blin`’ом можно пользоваться как обычным дистрибутивом. Доустановить софт при надобности можно при помощи программ из пакета `rpm`.



Рис.6

Дополнительно в будущем, возможно, планируется использование системы пакаджей наподобие FreeBSD – сайт медленный, и мне удалось попасть не во все закрома, поэтому точно сказать не могу, но в пользу данного предположения гласит наличие скрипта `blinpkgview` (Рис. 6) для просмотра установленных в системе приложений, внутри которого зарезервированы переменные вроде `BUILD`.

Итак, вывод. А почему бы и не blin? Он достаточно богат функционально, чтобы с него начать знакомство с Linux, его также можно с успехом использовать в повседневной деятельности в качестве обычной десктоп-системы (приложения вроде OpenOffice при желании можно доустановить и самим). И конечно же, blin неплох для настройки и исследования сетей и, естественно, серфинга по мировой паутине. К тому же его всегда можно таскать с собой. Как мне кажется, blin все-таки вышел не комом. И еще радует то, что, судя по всему, это самостоятельная разработка, с оригинальными инструментами и придумками, а не русифицированный зарубежный дистрибутив. Так что мы тоже умеем. Так держать!

Blin&Linux forever!

## Братство браузеров

Сергей А. ЯРЕМЧУК, grinder@ua.fm

*Так уже сложилось, что в Linux нет своего доминирующего веб-браузера, который встраивается в ядро, так или иначе навязывая себя пользователю. Изначально в системе присутствует несколько браузеров, и есть возможность выбрать любой исходя из эстетических соображений, личных пристрастий, задач – и, что самое существенное, наличия свободных системных ресурсов.*

Можно, конечно, спорить, какой из браузеров в Windows лучше, какой быстрее, но за время моей работы в этой системе я, честно говоря, не нашел того довода, который бы однозначно гласил в пользу того или иного браузера. Системе все равно приходилось тащить IE, который можно было вырезать только с помощью программы Lite98, но так как без родного браузера в Windows все равно не обойтись, полностью удалять его все же не рекомендуется. Netscape старых версий был примерно такой же по тяжести, но однозначно проигрывал в скорости и в качестве отображения страниц. А Opera иногда их так коверкала, что те доли секунд, на которые она их быстрее загружала, не казались таким уже выигрышем, да и с кириллицей она подружилась совсем недавно.

При первом моем знакомстве с Linux я тоже не нашел ни одного браузера, которым бы остался доволен на все сто. Но сейчас ситуация сильно отличается. Итак, давайте разберемся, что же за браузеры предлагаются пользователю для работы, тем более что в последнее время здесь замечено некоторое оживление. Для теста я использовал довольно сложную по структуре страницу с сайта <http://linux.tucows.com>, поэтому все недостатки того или иного продукта сразу же всплывали на поверхность.

## Консольные браузеры

Итак, Lynx (Рис.1) – легкий, я бы даже сказал, сверхлегкий браузер, понимающий практически все имеющиеся стандарты, принятые в web'е на сегодняшний день. Он позволяет обращаться к документам, находящимся в кэше, но неправильно отображает фреймы и таблицы. Вместе с тем, Lynx вполне пригоден, особенно на слабых машинах. Я, например, с его помощью при отладке просматриваю результат работы скриптов на Perl и PHP – быстро, и руки от клавиатуры отрывать не надо. Еще одна особенность выделяет данный браузер. Он особенно щепетильно относится к интерпретации кода, поэтому если есть серьезные изъяны, Lynx просто откажется работать. При запуске можно использовать дополнительные ключи, все они описаны в man. Я чаще всего использую -dump, выводящий как сам документ без работающих гиперссылок, так и отдельно все имеющиеся в нем гиперссылки. Это позволяет отобрать необходимые ссылки и загрузить их через wget. Для вывода исходного текста документа также нередко пользуюсь ключом -source.



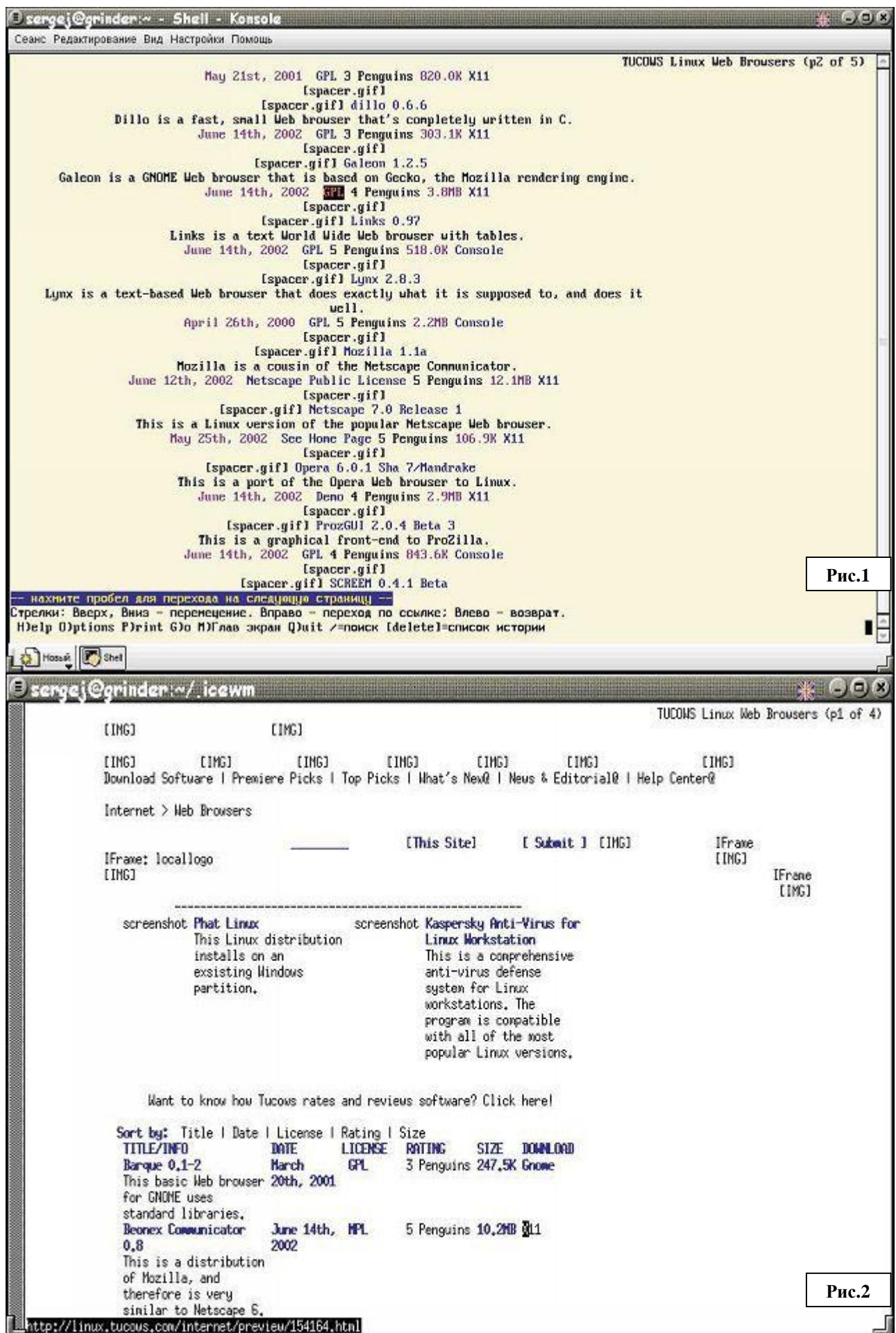


Рис.1

Рис.2

А вот **Links** (Рис.2) отлично справляется с выводом таблиц, в нем возможно использование меню, причем русифицированного. В остальном же Links практически не отличается от Lynx. Эти два браузера по традиции включены в каждый дистрибутив Linux.

Следующий браузер Zen 0.2.1 (<http://www.nocrew.org/software/zen>, 270 Кб, Рис.3) поистине многолик. Его изначальный замысел состоял в том, чтобы предоставить возможность пользователю выбирать по своему усмотрению интерфейс для работы. Программа построена по модульному принципу – часть, отвечающая за интерфейс, отделена от части, отвечающей за средства просмотра и отображения веб-страниц. В настоящее время доступны три интерфейса: zen – консольный вариант, в котором отсутствует возможность работы с гиперссылками, gzen – интерфейс на основе Gtk-библиотек, и fbzen – за основу взяты библиотеки oFBis (<http://osis.nocrew.org/ofbis>), которые вам понадобятся при установке. Но вот с выводом страниц у нашего подопечного пока не все в порядке, хотя при большом желании на простых сайтах проекта GNU его можно использовать.

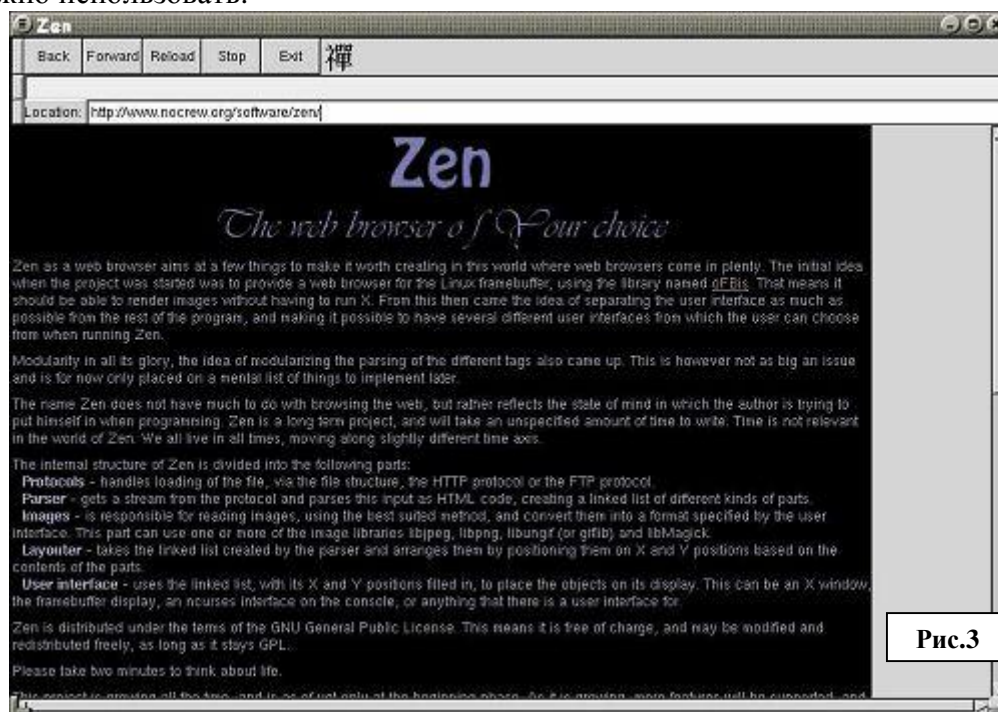


Рис.3

## Браузеры под X-Window

Начнем, пожалуй, с классики. **Netscape**, текущая версия 7 (<http://home.netscape.com>), размер дистрибутива 43 Мб, рис.4. Если раньше данный браузер только ругали, то в последнее время все чаще слышны похвалы. Интерфейс с множеством закладок кажется вполне удобным. И похоже, разработчикам удалось, наконец, избавиться от извечных проблем с отображением страниц и работой с CSS. Недостатком можно считать размер дистрибутива и его прожорливость по части использования системных ресурсов. Хочется напомнить, что в своем составе Netscape имеет также средство создания страниц Composer. И хоть я и не видел ни одного человека, сделавшего с его помощью сайт, подправить код страницы в Composer'е вполне возможно.

Но похоже, что в последнее время все силы разработчиков брошены на другой проект – Mozilla (<http://www.mozilla.org/build/unix.html>, рис.5), отделившийся от основного в 1998 году; все новинки сначала появляются в этом браузере, а потом уже переключиваются к старшему брату. В данном браузере воплощены все стандарты отображения страниц (как и в Netscape); возможно увеличение отображения web-страниц без потери качества (я пробовал – до 1000%), так что Oper'е хвастаться уже нечем; реализована поддержка плагинов; интегрирован Real Player, WinAmp и Net2Phone; предусмотрена возможность изменения внешнего вида с помощью тем (в стандартном комплекте их две – Classic и Modern, остальные доступны на сайте). Весом он поменьше (13.5 Мб), да и к системным ресурсам относится экономнее. Для начинающих web-программистов припасена небольшая подборка, знакомящая с некоторыми технологиями. Хочется напомнить, что



существует проект русификации Mozilla <http://www.mozilla.ru>, где можно найти необходимую информацию и файлы; дополнительно русский вариант можно найти на сайтах компаний AltLinux и ASPLinux.



Рис.4

И еще: в последних сообщениях, связанных с безопасностью, в Mozilla'х версий ниже 1.0.1 обнаружена уязвимость. И хоть она и не относится к разряду опасных, но если вы не хотите, чтобы ваше путешествие в Интернете кто-то мог отследить, обновите версию. В последнее время в адрес этого браузера в Интернете слышно столько похвал, что впору рассматривать его как серьезного конкурента IE. Пожалуй, тягаться с OpenSource, когда браузер создают не десяток программистов, а пару сотен, последнему будет трудновато.

Следующей на очереди будет **Opera 6.0.3** (рис.6), лишь недавно пришедшая в мир Linux. Домашняя страница <http://www.opera.com>, размер дистрибутива в зависимости от того, в каком виде вы будете его загружать и какие библиотеки требуются (статические или динамические), колеблется от 1.5 до 4.37 Мб. Как обычно, данный браузер не является бесплатным, за него на сайте требуют \$39 (и это для Linux!), иначе вам придется созерцать рекламный баннер. Нравится мне в нем многодокументный интерфейс – за это я его люблю использовать в оффлайне, а также возможность назначения псевдонима браузеру (т.е. можно выдать его за ту же Mozilla (5.0, 4.78, 3.0) или MSIE 5.0) и активации/деактивации режима отображения графики нажатием одной кнопки, «на лету». Также очень удобна реализованная пока только в этом браузере (в Mozilla и Netscape в скором времени появится) функция Mouse gestures, позволяющая производить основные операции с помощью определенного жеста мышкой (например, с нажатой правой кнопкой мыши двиньте ее сначала вниз, а затем без остановки влево и отпустите кнопку – окно минимизируется). Ну что я могу сказать – в Linux данный браузер не является самым быстрым, а вот системных ресурсов потребляет гораздо меньше предшествующих. Зато, в отличие от последних, за него дают всего четыре коровы, к тому же и платить надо.



Рис.5

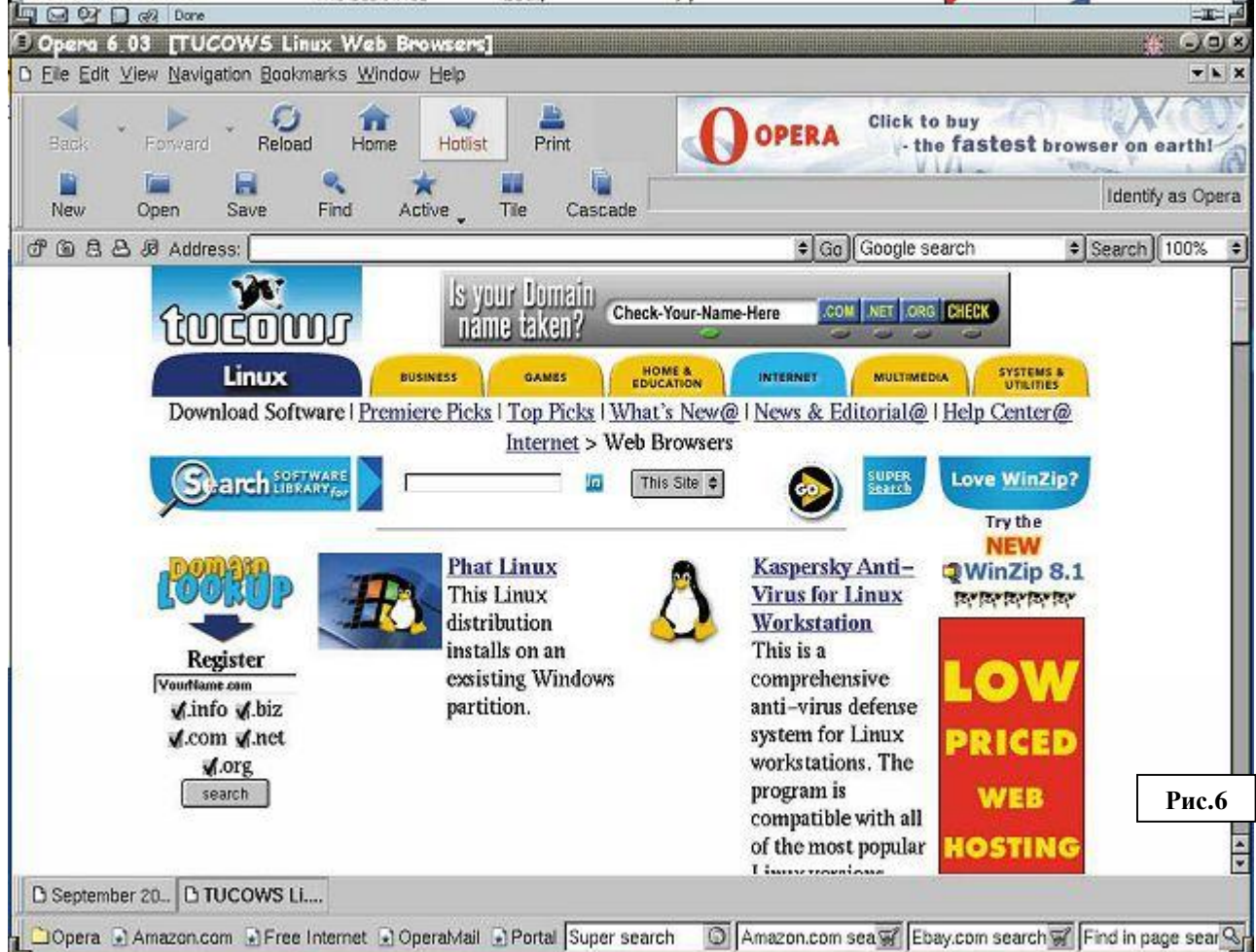


Рис.6



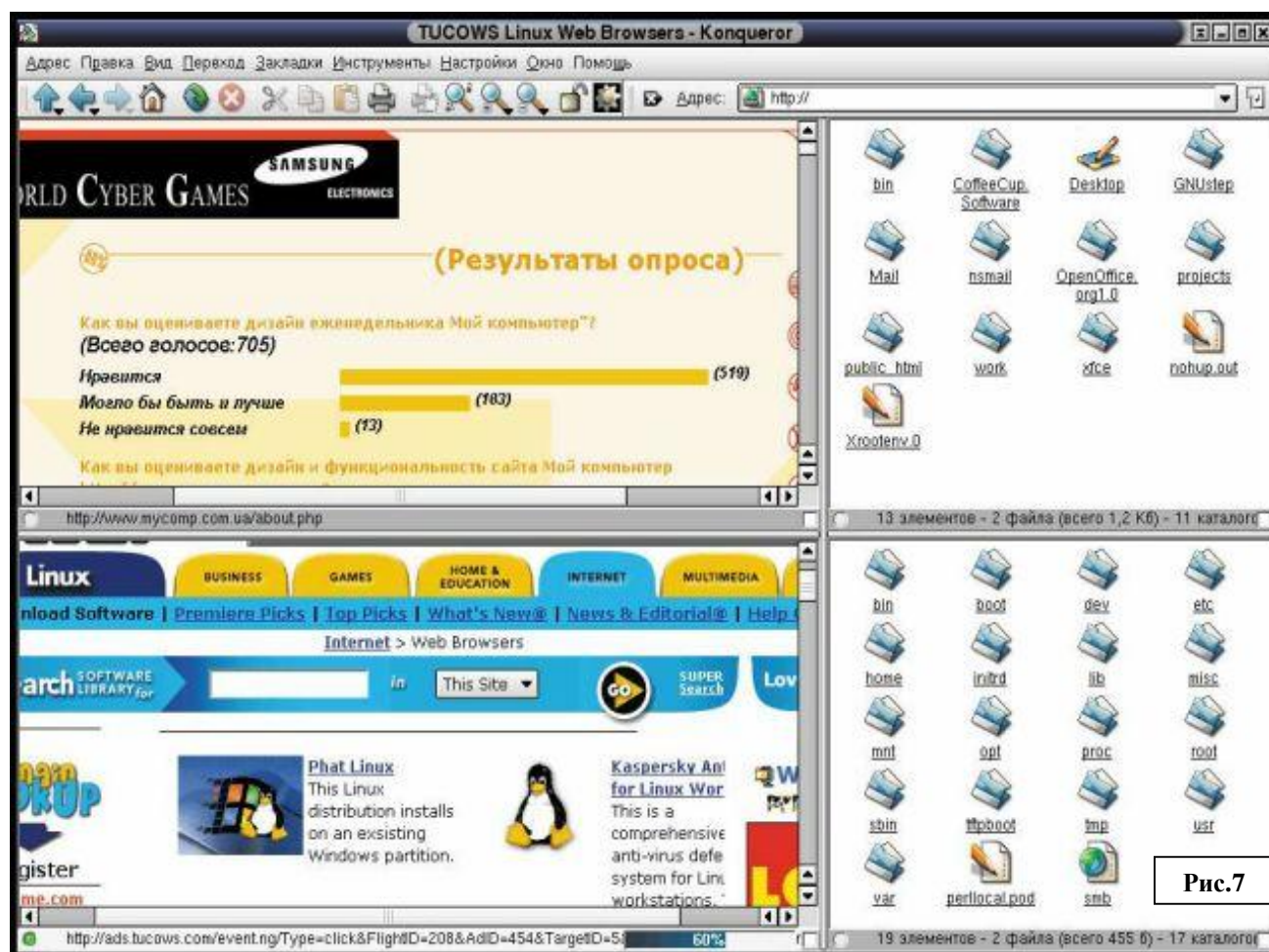


Рис.7

Теперь самое время вспомнить о **Konqueror** (<http://www.konqueror.org>), стандартно входящий в поставку KDE. Это и веб-браузер, и файловый менеджер в одном лице (не говоря уже о просмотре графики). Если раньше у него были проблемы с отображением веб-страниц, то теперь его можно считать полноценным браузером. При установке русского языка в качестве основного все пункты меню будут автоматически русифицированы, плюс реализована возможность импорта закладок из Netscape и возможность конфигурирования полностью по своему вкусу, включая внешний вид. Не забывайте также о предустановленных профилях, позволяющих превратить Konqueror и Midnight commander в удобное средство предпросмотра и в собственно web-браузер. Просматриваемую страницу можно переслать в любую внешнюю программу или сохранить в заархивированном виде. Особенно интересна возможность разделения главного окна на несколько независимых частей (рис.7). Так же, как и в Opera, можно назначить псевдоним браузеру, что решает проблемы с отображением страниц (а их я и так не нашел). Konqueror понимает плагины Netscape, которые могут быть импортированы в него; для наиболее часто посещаемых узлов можно создать сокращения для ввода; предусмотрен полный контроль над кэшем и над cookie – последние можно настроить конкретно для каждого узла. Поддерживаются все методы шифрования, какие можно встретить в Интернете, с возможностью их настройки и проверки правильности подписи. По части скорости отображения страниц находится где-то на уровне Opera. Что я могу сказать – любимчик, да и все тут.

### Браузеры с кодом Mozilla

Следующие два браузера основаны на коде Mozilla; в них создатели убрали лишнее, попытались сделать дружелюбными к пользователю и снизить потребление системных ресурсов. Так что если ваш любимый браузер – Mozilla или Netscape, но по причине их прожорливости они не совсем вам подходят, обратите внимание на приведенные ниже браузеры.



Рис.8

Первый, **Beonex 0.8.1** (рис.8), имеющий дополнительно код, специфичный для Netscape. Домашняя страница <http://www.beonex.com>, размер дистрибутива соизмерим с Mozilla – 10.7 Мб. Разработчики его считают, что Mozilla не предназначен для пользователей и сильно громоздок, а потому они просто не могли не создать beonex. Распространяется уже скомпилированным под разные типы процессоров, поэтому будьте внимательны при загрузке. Установка сводится попросту к распаковке архива. Имеет в своем составе: Navigator – веб-браузер, Mailnews – e-mail клиент и средство просмотра новостей, Composer- средство создания страниц, а также ChatZilla и адресную книгу, плюс хороший встроенный менеджер загрузок. В остальном сильно напоминает своих родителей (даже по внешнему виду сразу не отличишь), в том числе и по части работы с русскими кодировками, но существенно легче и быстрее, что особенно чувствуется на моем домашнем компьютере.

Следующий – **Phoenix 0.2** (рис.9), <http://www.mozilla.org/projects/phoenix>, размер дистрибутива 11.1 Мб. Браузер, подобный Galeon, K-Meleon и Chimera, но полностью написан при помощи XUL и спроектирован так, чтобы использоваться на многих платформах. Отличительной чертой этого браузера является то, что разработчики концентрируют свои силы не на внешних особенностях и удобствах (интерфейс не должен быть громоздким и отвлекать от работы), а на скорости работы. Остальными особенностями данного браузера являются: легко настраиваемый toolbar (в том числе и возможность создания персонального), хорошая скорость отображения страниц (на сайте заявлена в 2 раза большая по сравнению с Mozilla – это не совсем так, но очень близко), удобная конфигурация закладок и истории посещения страниц, удобство работы со списками, довольно неплохой менеджер загрузок, поддержка различных плагинов и аддонов, включая темы для изменения внешнего вида (как от Mozilla, так и свои собственные – <http://themes.mozdev.org/phoenix>), поддерживает кириллицу с возможностью установки шрифтов. Кстати, данный браузер доступен и для Windows-платформы (<http://ftp.mozilla.org/pub/phoenix/releases/0.4/phoenix-0.4-win32.zip>), так что пользователям этой системы советуем обратить на него внимание (не работает на Windows 95), достойный конкурент



Орег'ы. Да, пока я тут писал статью, вышел следующий релиз Phoenix 0.4 (Oceano). В котором многочисленные улучшения коснулись: реакции на всплывающие окна; работы с панелью инструментов (перемещение, добавление новых элементов) и горячими клавишами; появились вкладки (как в Mozill'e). Так что спешите, пока не выпустили версию 0.5. Это единственный известный мне браузер, у которого с увеличением номера версии (а значит, и функциональности) объем дистрибутива, наоборот, уменьшается: в версии 0.4 – уже 9.5 Мб, а вообще планируется остановиться где-то на отметке 6 Мб для Windows и 7 Мб для Linux.



Рис.9

### Экономичные браузеры

Следующие браузеры имеют только базовые функции и настройки, а также упрощенный интерфейс (некоторые из них даже не позволяют сохранить просматриваемую страницу), но зато занимают гораздо меньше ресурсов, да и скорость отображения страниц выше.

**Barque** (<http://barque.sourceforge.net>, 250 Кб, рис.10) – браузер, основанный на гномьих библиотеках, использующий GtkHtml и поддерживающий технологию Vonobo. В настоящее время поддерживает только протокол HTTP, так что просматривать с его помощью файлы на локальных дисках вряд ли получится. Не поддерживает загрузку файлов с помощью FTP и HTTP и совсем не понимает фреймов, русского языка и протокола HTTPS, позволяющего устанавливать защищенные соединения. Но что вы хотите от версии 0.1?

**Cheetah 0.10** (<http://cheetah.sourceforge.net>, 180 Кб, рис.11) – еще одна попытка сделать Mozill'у легче. Этот браузер позволяет быстро перемещаться по просторам Интернета, загружать файлы с помощью HTTP- и FTP-протоколов. Собственно, вот и все. Авторы утверждают, что для остального у них есть другие программы, специально для этого предназначенные. Не обошлось и без изъянов: русской кодировки не понимает, визуальных средств для ее настройки нет (и даже help'a никакого нет), сильно искажает сложные страницы (даже свою домашнюю не может нормально передать), не понимает основных графических форматов файлов (.gif, .png), но на сайтах проекта GNU придется в самый раз.

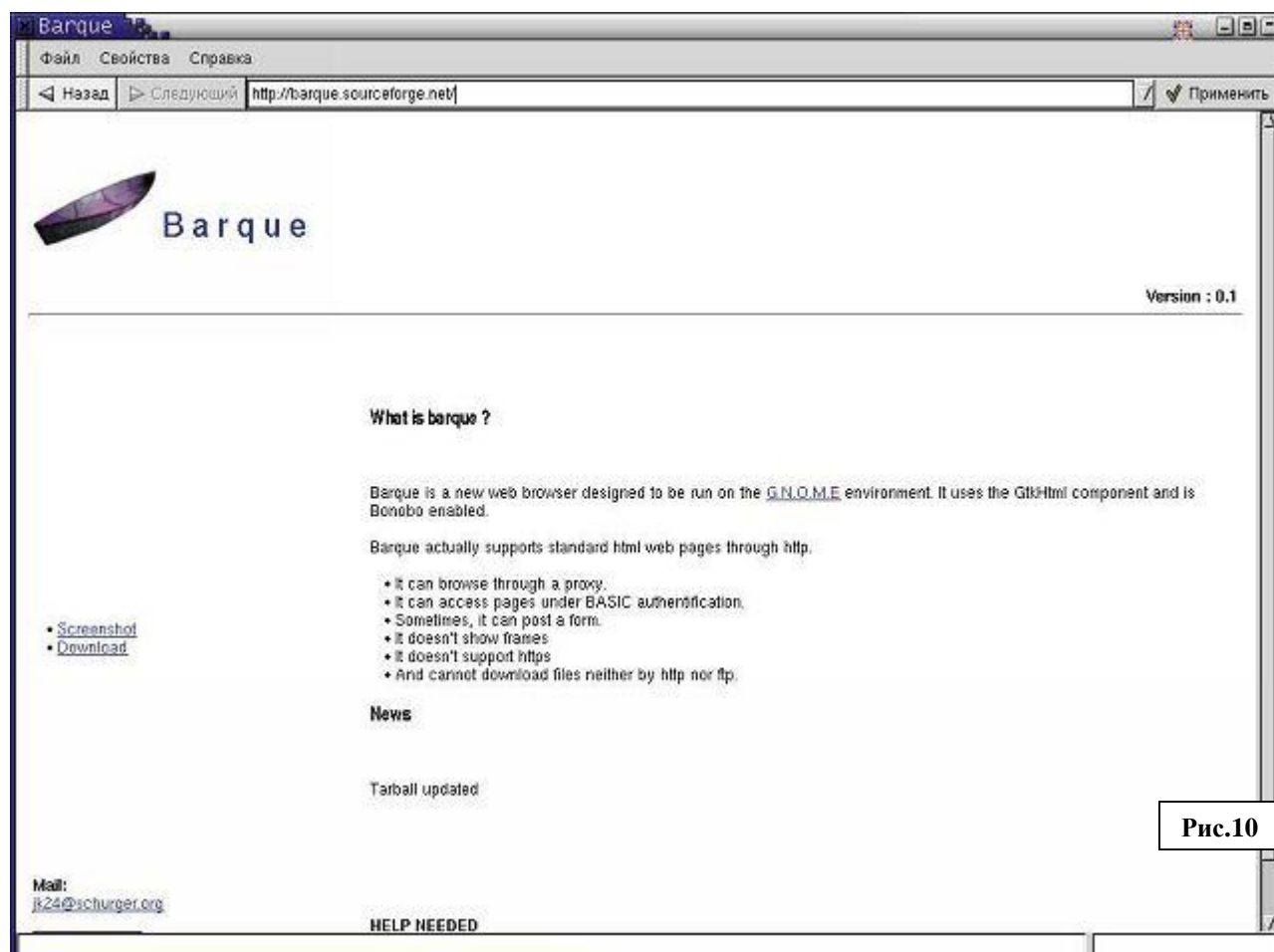


Рис.10

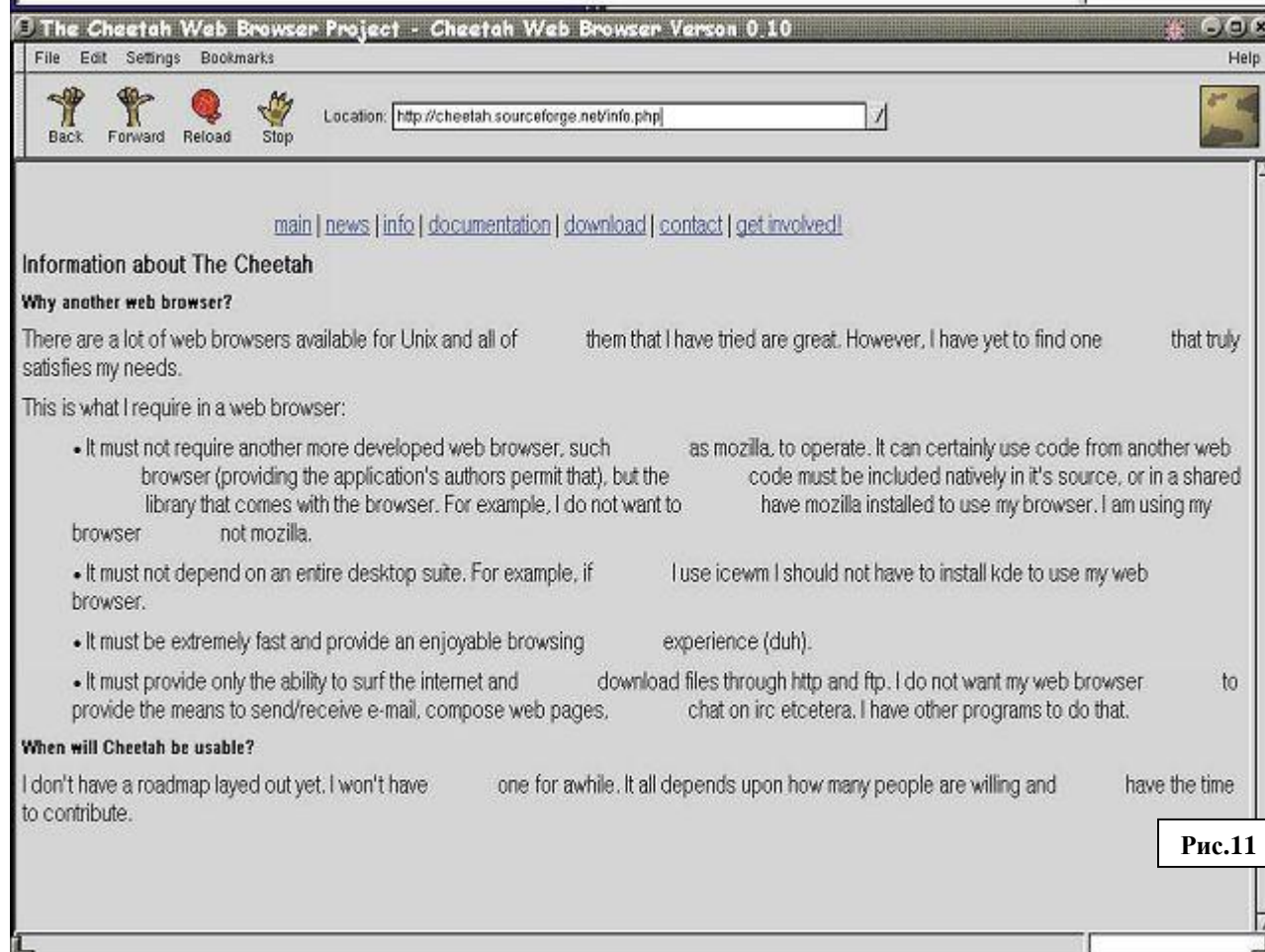


Рис.11

**Dillo 0.6.6** (<http://dillo.cipsga.org.br>, 300 Кб, рис.12) – а вот это уже серьезно. Данный проект развивался в последнее время просто бешеными темпами. Браузер написан на языке С и практически с нуля, для своей работы требует библиотеки Gtk+, при этом необязательно наличие установленного Gnome; для анализа страниц использует переработанный анализатор gzilla (угадайте, откуда). Позволяет работать с cookie, поддерживает SSL. В ближайшее время планируется портирование под библиотеку Gtk+ 2 и доведение до ума работы с протоколом FTP и фреймами. По скорости обработки страниц далеко обходит «the fastest browser on earth» — Opera. Список поддерживаемых платформ просто впечатляет, здесь кроме всевозможных реализаций Linux и BSD-систем, а также Solaris, можно встретить MacOS и, что особенно в редкость, таких как iPAQ, Psion, QNX и даже SONY PlayStation2. Огорчает то, что по умолчанию не отображает символы кириллицы, и для настройки придется повозиться. Единственным дистрибутивом, в котором можно сейчас встретить патченный Dillo, является Multilinux (<http://multilinux.sakh.com>). Для индивидуальной настройки параметров скопируйте файл /usr/local/etc/dillorc в каталог \$HOME/.dillo.

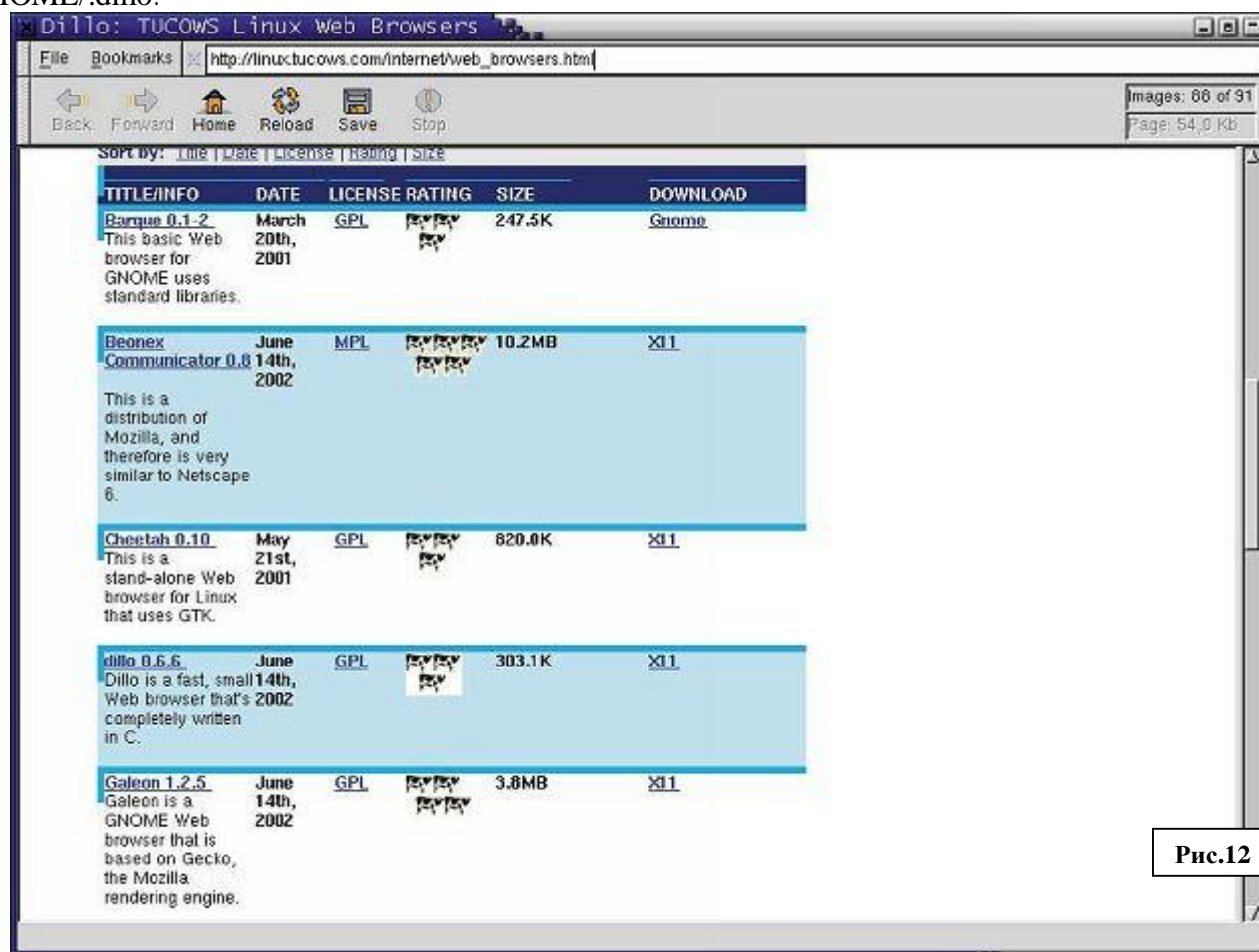


Рис.12

**Skipstone 0.8** (<http://www.muhri.net/skipstone>, 320 Кб, рис.13) – еще один браузер, основанный на библиотеке Gtk+, использующий компонент Gecko, применяемый в Mozilla; построен по модульному принципу. Для загрузки файлов использует консольную утилиту skipdownload, ведущую свою родословную от wget (ее можно использовать и отдельно); для работы с почтой можно подключить соответствующую программу. Если при запуске программы возникнут трудности, откройте в каком-либо текстовом редакторе скрипт skipstone, находящийся приблизительно в /usr/local/bin, и установите значения переменных LD\_LIBRARY\_PATH и MOZILLA\_FIVE\_HOME, ссылающиеся на каталог с Mozilla.

У меня хоть и был указан в одном из вариантов правильный путь, но браузер наотрез отказался запускаться, жалуюсь, что не находит библиотеки, — тогда я просто закомментировал лишнее и привел соответствующий фрагмент скрипта к такому виду:



```

export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/usr/local/mozilla
export MOZILLA_FIVE_HOME=/usr/local/mozilla
if [ -f ./skipstone-bin ]; then
exec ./skipstone-bin $@
else
exec skipstone-bin $@
fi

```

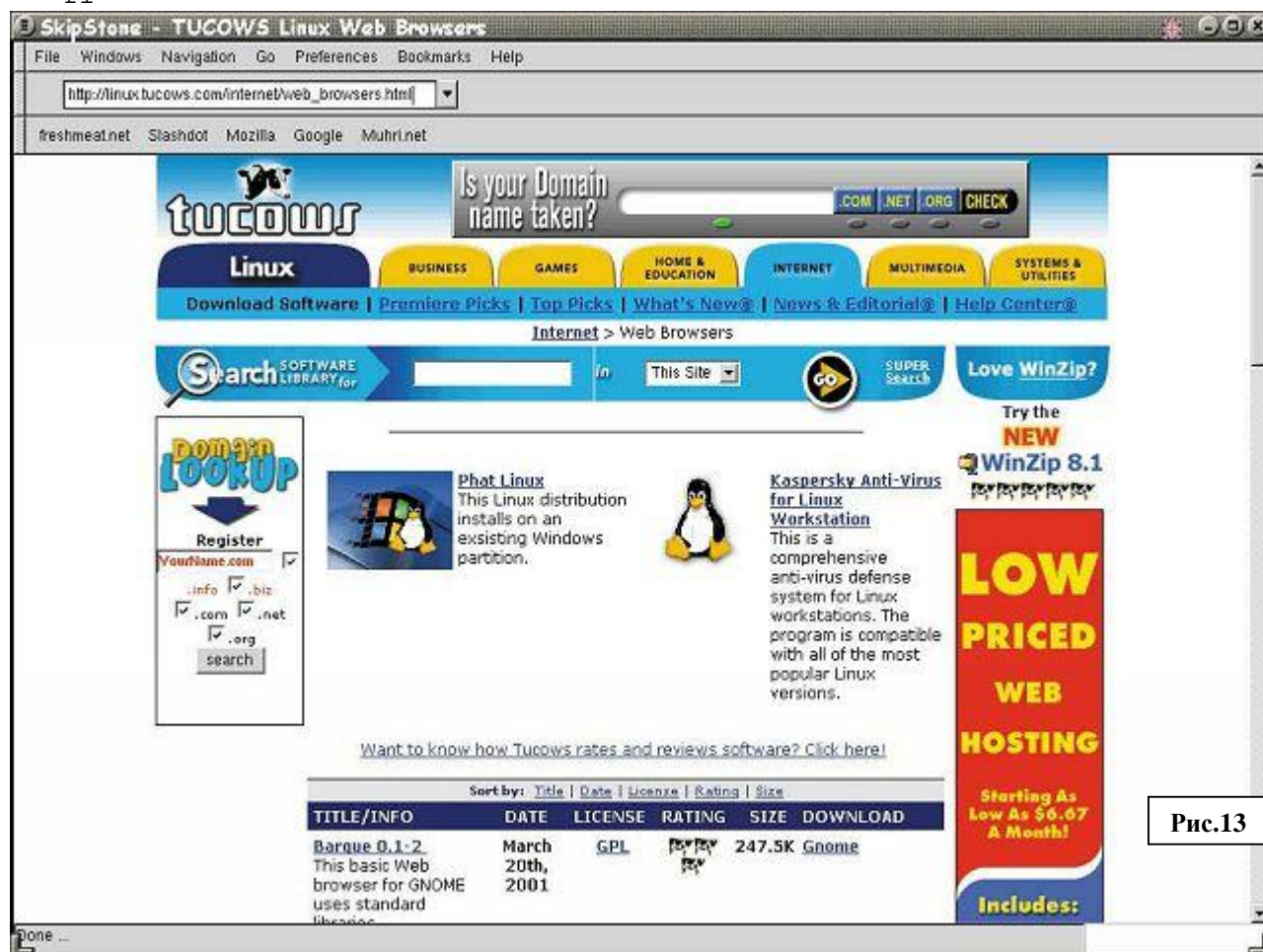


Рис.13

Это основной конкурент dillo. По загрузке системы и скорости обработки файлов они находятся примерно на одном уровне, но, как уже говорилось, skipstone имеет отличный загрузчик, к тому же изначально поддерживает кириллицу во всевозможных кодировках с возможностью установки шрифтов, позволяет сохранять сессии, просматривать исходный код страницы, имеет удобные менеджеры истории посещения и закладок, настраиваемый toolbar, поддержку тем. Так что, сложив все вместе, я бы отдал именно ему первое место в анкете МК, результаты которой, скорее всего, уже будут известны к моменту выхода статьи. Вот и все, что мне удалось накопать. Впрочем, есть и другие браузеры для платформы Linux. Об одном из таких – удобном и быстром браузере Galeon – я уже рассказывал на страницах журнала, советую не упускать его из виду. Как видите, количество программ, с помощью которых можно создать web-страницу, гораздо меньше средств их просмотра. Это показывает, что Linux уже нельзя считать системой, рассчитанной только на разработчиков. Судя по возрастающему количеству писем, приходящих на мой адрес, количество пользователей Linux растет с каждым днем. Мне уже трудно справиться с этим напором. Но как говорится, «одна голова хорошо, а две лучше». Поэтому хочу посоветовать несколько «горячих» форумов по теме, где можете дополнительно поискать ответ на ваш вопрос. Мой самый любимый – на OSZone.net (<http://forum.oszone.net>), а вот и остальные: <http://www.linuxoid.ru/phorum>, <http://linuxnews.ru/forum>, зайдите также на сайт NoMicrosoft <http://forum.nomicrosoft.ru/pbb/index.php>. И последняя ссылка — сайт совершенно нового журнала Системный Администратор, в сентябре вышел первый номер – <http://www.samag.ru>. Основная тема журнала – Unix; здесь можно встретить любые вопросы,

начиная от работы системы до программирования (последние «горячие» темы: сетевые нюхачи и сканеры портов). И хотя сайт еще молодой, но быстро обрывает публичкой. На сегодня, пожалуй, все.